UNCLASSIFIED

AD NUMBER
ADB021192
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; AUG 1977. Other requests shall be referred to Rome Air Development Center, Griffiss AFB, NY3441.**
AUTHORITY
RADC ltr 14 Apr 1980

THIS PAGE IS UNCLASSIFIED

ADB021192

(2)

RADC-TR-77-247
Final Technical Report
August 1977

FTD USER COMMUNICATIONS INTERFACE

Planning Research Corporation

Distribution limited to U. S. Gov't agencies only; test and evaluation; August 1977. Other requests for this document must be referred to RADC (IRDT), Griffiss AFB NY 13441.

AD No.____
DDC FILE COPY

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed and is approved for publication.

APPROVED: Roberti N. Ruberti

ROBERT N. RUBERTI
Project Engineer

APPROVED:

ROSS H. ROGERS, Colonel, USAF
Chief, Intelligence Reconnaissance Division

FOR THE COMMANDDR: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL
1776-1976

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-77-247	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): FTD USER COMMUNICATIONS INTERFACE	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, June 1975 - April 1977
	6. PERFORMING ORG. REPORT NUMBER: PRC-R77-011
7. AUTHOR(s): N/A	8. CONTRACT OR GRANT NUMBER(s): F30602-75-C-0345
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Planning Research Corporation, 7600 Old Springhouse Road, McLean VA 22101	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 64750F, 20530201
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRDT), Griffiss AFB NY 13441	12. REPORT DATE: August 1977
	13. NUMBER OF PAGES: 86
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
110p.	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Distribution limited to U. S. Gov't agencies only; test and evaluation; August 1977. Other requests for this document must be referred to RADC (IRDT), Griffiss AFB NY 13441.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Robert N. Ruberti (IRDT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Intelligence Data Handling
User-oriented language design
Interactive data analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A design specification and implementation plan for an integrated user communication interface for FTD has been developed. The interface will provide analysts with the flexibility to interactively define functions which query the data base and perform operations on data extracted from the data base without the intervention of a programmer.

DD FORM 1473, 1 JAN 73 — EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

402 560

4B

TABLE OF CONTENTS

EVALUATION

The User Communication interface designed in this effort is being developed to provide FTD a centralized facility for accessing data bases, executing special intelligence application programs and generating preliminary and finished intelligence reports. This effort is included in TPO Thrust R3D.

Robert N. Ruberti

ROBERT N. RUBERTI
Project Engineer

1.0 INTRODUCTION AND SUMMARY

This volume has been written for the FTD analyst community as an introduction to the User Communications capability. It discusses the motivation for such a capability, the software and hardware design philosophy, and User Communications evolution in the light of FTD's near term and long range ADP planning (Sections 1 and 2).

In Section 3, which describes User Communications from the user's point of view, the reader is taken on a guided tour of the proposed user station, and the function of each hardware component of the station is explained (Section 3.1). Section 3.2 illustrates the User Communications language in scenario form.

Section 4 concentrates on the User Communications language and software capabilities from a systems point of view, explaining how the system was designed with attention to human cognitive processing faculties, emphasizing the relationship of graphics to certain types of cognition and the role of special user structures which will assist the FTD analyst in planning a task, and will provide him with an 'audit trail' of his own activities (4.1.1: The Man-Machine Interface).

Critical design decisions, emerging from the need for language support of non-computer oriented users as well as analysts who are competent programmers, are reviewed in Section 4.2.3 so that users may become aware of design tradeoffs involving the range of user skills and backgrounds. The functional structure of the user language is outlined in order to illustrate how it can be expanded in one direction to provide a more

computer oriented language capability (e.g., an APL-like language), and in another, to provide an English-like user language (4.2.2). These features, among others, will allow access of a subset of User Communications capabilities via standard terminals now in use in the FTD environment.

Section 4.2.4 addresses the motiviation for a User Communications breadboard system and user hands-on evaluation of the breadboard in the FTD environment.

Appendix A contains a description of a baseline hardware configuration for one user station, with estimated costs for the various components.

As mentioned above, this technical report is intended as an introduction to User Communications for the FTD analyst community. Detailed information on the system is contained in References 1, 2, 3, and 4.

1.1 Why User Communications

The primary mission of the Foreign Technology Division of the Air Force Systems Command involves the acquisition and analysis of data received from a variety of sources -- including electronic signals, imagery, and text -- and the synthesis of these data into scientific and technical intelligence which can be used as a basis for strategic and tactical decisions. The results of these extensive and detailed analytic and synthetic processes are finished intelligence reports satisfying long term and special case DoD information requirements. These constitute the information product of FTD; thus, the quality of the product and the productivity of the analyst/information producer are critical factors in the performance of FTD's mission.

As the consumers of FTD Scientific and Technical (S&T) intelligence products and services grow ever more numerous, the demand for FTD products and services rapidly increases, inevitably causing a rapidly increasing workload for the FTD activity.

The current FTD challenge is thus to increase intelligence production in order to meet the increased workload. One possible response to this challenge is automation of some processing steps in the analytic and information manipulating tasks of analysts, the objective being to increase analyst productivity by offloading appropriate functions to computers.

In order to achieve effective exploitation of automated resources, however, two problems must be overcome. These can be characterized as:

- <u>the machine problem</u> -- that of making sufficient computing resources available to potential analyst users;
- <u>the human problem</u> -- that of motivating analysts to utilize computers.

FTD is currently considering an ADPE plan which would provide a solution to the machine problem. Moreover, in their sponsorship of the User Communications development effort, RADC and FTD are pursuing a solution to the more subtle problem of motivating non-computer-oriented analysts to utilize automated support.

1.2 FTD User Communications Design Features

User Communications design features which will insure a 'friendly' interface include:

- o system-generated tutorials to explain User Communications features and operation;
- o continuous prompting to assist user input;
- o use of graphics to provide an image of complex relations in the STIS data base;
- o verbose error diagnostics and 'help' functions.

On the other hand, the experienced user -- who does not require extensive tutorials and system messages to perform his tasks -- may operatre in a terse mode, which limits communications to the essentials.

The user language is functionally oriented, being designed to operate in a manner similar to a hand calculator. Like the calculator, it has keys for various operations; but since the repertoire of functions is much more extensive and varied, a special function keyboard is provided (see Section 3.1.2 for a detailed description of the function keyboard). The keyboard is organized to provide three major resources for users:

- o Keys for (system-provided) user language functions;
- o Keys for user memories;
- o Keys that are initially unassigned -- reserved for new functions to be built up by the user out of existing functions.

The first group function keys is the heart of the user language; each of these activates one of the language commands. The second group of keys corresponds to locations provided by the system for temporary storage of results of user computations -- these correspond to the memories provided in the more sophisticated hand calculators.

One of the virtues of the system design is that users can expand the user language by composing their own functions. The DEFINE_FUNCTION command causes the system to accept a sequence of user language commands which can have the effect of providing a new language capability. The user then assigns his new function to one of the unused function keys (the third group of keys mentioned above), and it becomes, temporarily or permanently, a part of his user language (for each user the system will know about user-defined functions and key assignments). This is equivalent to adding a new function to a hand calculator. Examples of user-defined functions are given in Section 3.2.1.

The level of (system defined) user command functions contains:

- data manipulation functions ("RETRIEVE_VALUE");
- mathematical and logical functions ("ADD", "GREATER_THAN OR_EQUAL_TO");
- simple graphics functions ("PLOT");
- text editing functions ("DELETE_WORD");
- report generation functions ("SKIPLINE");
- special operators ("DEFINE_FUNCTION").

For each function (i.e., command) in the user language, there is a corresponding physical key on a keyboard, so that depressing the key on the function keyboard causes the user language function to be entered into the execution stream. The key is labelled, and is thus identified with the function name. The only actual typing required by the user (i.e., from the typewriter keyboard) is the variables (i.e., the numbers or text strings) that are the parameters to user language functions. Where it is appropriate, functions may also be selected from graphics "menus", using the data tablet and pencil (See Section 3.2.1).

The "DEFINE_FUNCTION" command is a special operator which allows the user to write his own functions by a sequence of key pushes and entry of variables. Once he has defined a function the user may assign it to an usused key on the function keyboard. This command provides the extensibility feature which permits a user to expand the repertoire of system functions to accommodate his particular analytic and information processing tasks.

Adaptivity to individual users and dynamic shifts in user tasks is provided by a transaction monitor which generates feedback data for system improvement and modification.

In addition to these user-oriented design features, the FTD User Communications system has software and hardware design features which provide a high degree of machine independence. The system design has been considered in terms of compatibility with the proposed FTD ADP plan. The basic replicable hardware configuration is now envisioned as a PDP-11/70 user station supporting from 1 to 20 users who interact locally and with the

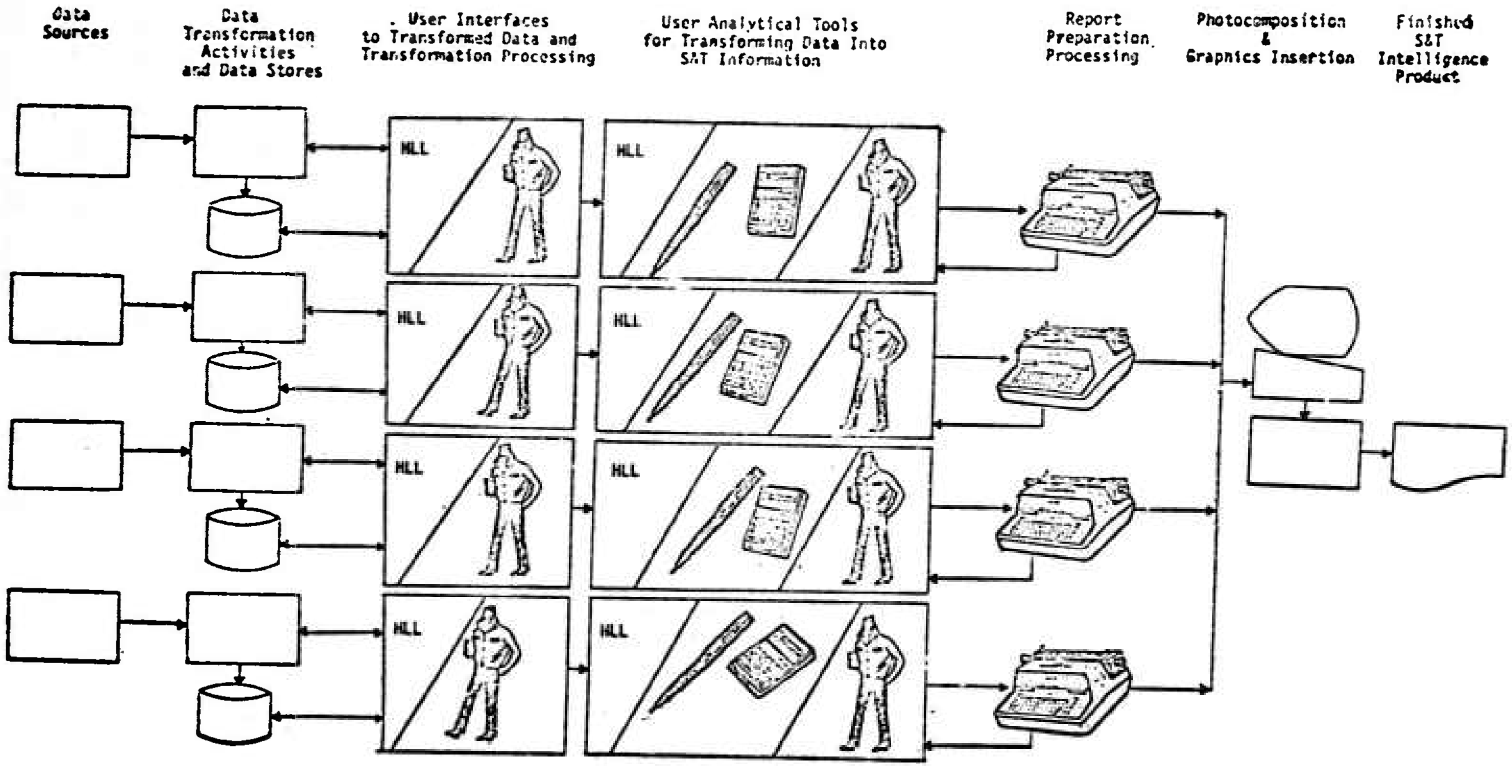


Figure 1-1. Production of Scientific and Technical Intelligence at FTD: Current Methodology

FTD Univac 1110 host data base machine via textual and graphic CRTs. A baseline configuration for the User Communications breadboard system would include the following items:

- typewriter input keyboard;
- CRT display for typed information and system responses;
- hard copy device for text;
- CRT graphics display (refresh type) with display processor and refresh memory;
- hard copy device (plotter) for graphics;
- graphics tablet and pencil;
- local on-line storage (e.g., disk) for text and graphics.

The PDP-11 based user station has the advantage of off-loading the heavily burdened U1110, and providing a system which will not require replacement when the U1110 is replaced in 1980.

1.3 The FTD Production Environment With and Without User Communications

1.3.1 Without User Communications: Current Production Methodology. At present, as shown in Figure 1-1, the varieties of data sources feeding into FTD are subject to a variety of preprocessing and processing operations to transform raw input into data useful for analysis. The output of these processes results in a variety of data stores, as shown in the second column of the figure. Unfortunately, analyst users must access these various data stores and call the many processing programs via a bewildering variety of protocols and query formats, unique to each data transformation or reduction activity. As shown in the third column, the analyst who is

fortunate enough to be also a programmer can attempt to deal with the various systems by writing his own access programs in a higher level language such as FORTRAN or COBOL. The non-computer oriented user requires the services of a programmer (indicated by the human figure) to access these data and processing algorithms. The result is inevitably a proliferation of user interface programs for accessing specific data and processing algorithms. Currently, such programs exist or are under development for IPS and IEAS users, and, if the trend of replicating the development of special user interfaces for each new program continues, the difficulties which users now experience in attempting to comprehend the number of access protocols, query formats, etc., can only intensify, resulting in increasing user frustration with the FTD computing environment.

The same situation exists with respect to user analytical tools for transforming data into scientific and technical information. As shown in the fourth column of Figure 1-1, the programmer-user has the advantage of performing at least some of his analysis by developing his own analytical algorithms in a higher level programming language such as FORTRAN. However, the non-computer oriented user is left with pen and calculator to analyze his data, unless he can obtain the services of a programmer to write algorithms for him. Again, this results in a proliferation of special case analytical programs, which are only of use to one user group, perhaps for a very limited period of time.

The next step in the generation of an FTD S&T report seems even more frustrating to users, as much repetitive typing and consequent proof and review is involved in the actual clerical preparation of the report

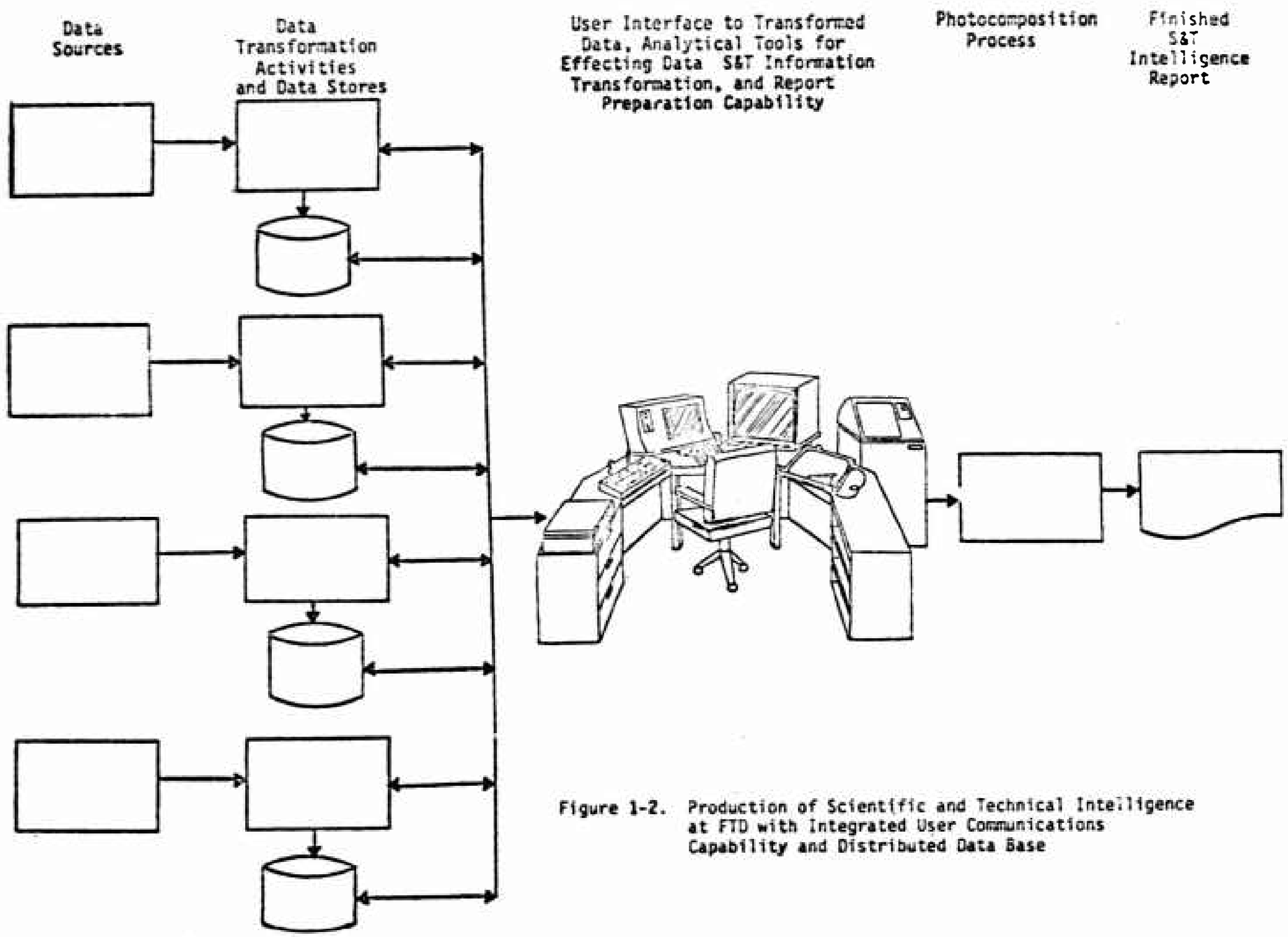


Figure 1-2. Production of Scientific and Technical Intelligence at FTD with Integrated User Communications Capability and Distributed Data Base

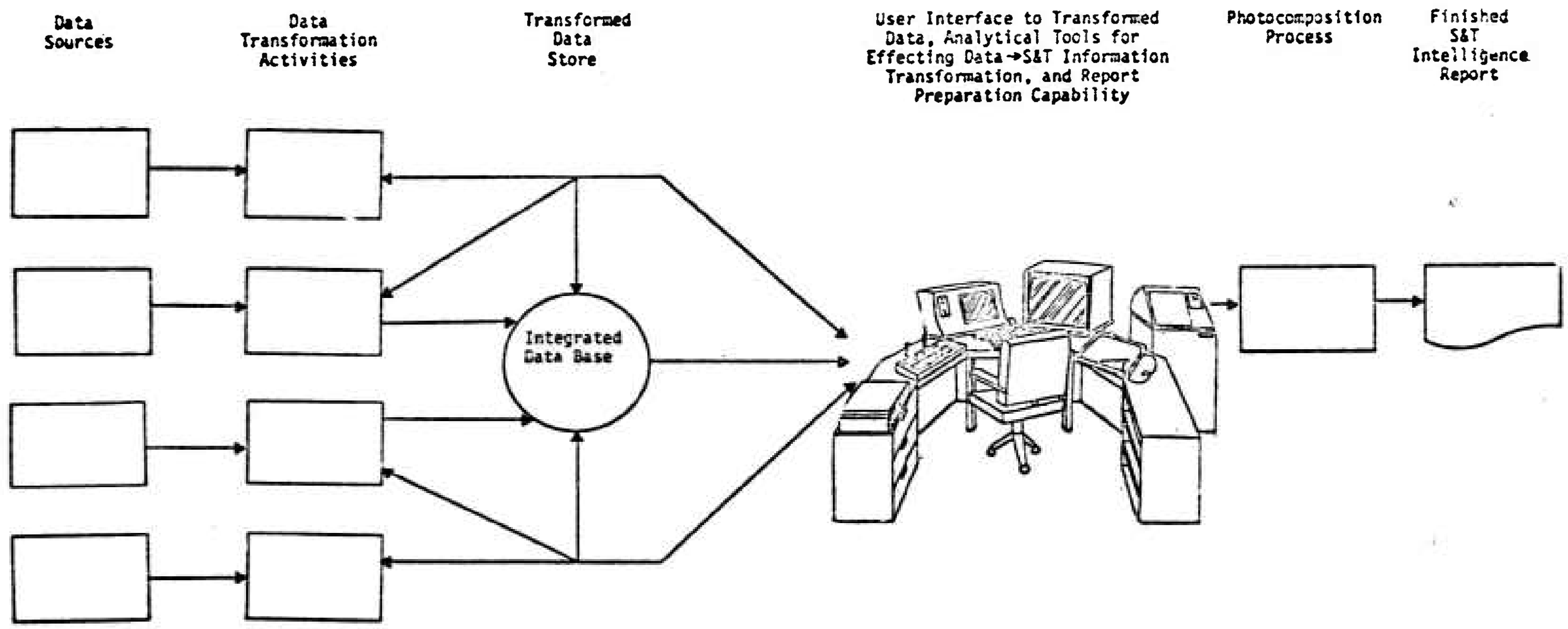


Figure 1-3. Production of Scientific and Technical Intelligence at FTD with Integrated User Communications Capability and Data Base

prior to the photocomposition and graphics insertion step, which outputs the final product. As much as nine months of clerical effort may be required to produce a study generated by an analyst in two months -- an elapsed time of almost a year between acknowledgement of an externally generated information requirement and production of a report satisfying that requirement.

1.3.2 The FTD Production Environment with User Communications: Proposed Production Methodology. The purpose of the FTD User Communications capability is to eliminate the fragmentation of efforts and proliferation of special user interfaces and user analytical tools by providing an integrated interface for communicating with all FTD automated systems and computerized data bases through a single medium, as shown in Figure 1-2. This also serves as an automated workbench for the non-computer oriented analyst, who can substitute an intelligent terminal and user defined functions for pen and calculator, inserting himself (or his secretary) into the report generation process with sophisticated text editing functions. The text and graphics he generates at the terminal will of course be in machine-readable form, eliminating the necessity for rekeyboarding for the photocomposition process.

As shown in Figure 1-3, the User Communications capability can utilize an integrated data base, or distributed data bases, as shown in the preceding figures.

2.0 TECHNICAL DISCUSSION

As stated in the preceding section, FTD's basic mission consists in the production of accurate and timely intelligence relating to foreign science and technology. With the advent of more sophisticated collection systems, the FTD analyst must sift increasing amounts of information from a variety of sources in order to produce his intelligence estimates of foreign technological capabilities and limitations. At the same time, increasingly many requests for scientific and technical intelligence products involving both quick reaction and longer range production tasks result in a rapidly growing workload.

In order to meet the challenge of increasing all source data volumes and increased demand for products and services, FTD has initiated a strategy of offloading analytical and information processing functions to computers with the ultimate objective of increasing analyst productivity while maintaining the quality of FTD intelligence products.

The expansion of automated support for analyst activities is inevitably accompanied by two major problems, identified above as:

- the machine problem -- that of making computing power available to analysts;
- the human problem -- that of motivating analysts to utilize computers.

As a solution to the first problem area, FTD has initiated the definition of an ADPE plan which, if implemented, could insure the availability of computer resources throughout the FTD analyst environment.

The solution to the problem of user motivation for exploitation of automated support is embodied in the proposed User Communications capability.

This section discusses the ramifications of these fundamental assumptions relating to the proposed development of a system for FTD User Communications. Section 2.1 describes the FTD mission and organization, Section 2.2 provides a background discussion of the evolution of the proposed User Communications concept, Section 2.3 explores the impact of some preliminary concepts for the FTD ADP 83 Plan, and Section 2.4 presents the User Communications design philosophy and a brief scenario illustrating interaction with the proposed User Communications breadboard system.

2.1 FTD Mission and Organization

The overall mission of Headquarters, Foreign Technology Division is to acquire, collect, analyze, produce, and disseminate foreign aerospace scientific and technical (S&T) intelligence and intelligence information; conduct an integrated analysis program; operate an S&T intelligence data handling system; collaborate with other organizations to improve the collection, acquisition, and utilization of foreign technology and intelligence; and develop and maintain the highest attainable level of knowledge concerning foreign aerospace technology, capabilities, and

limitations. Toward accomplishment of this mission, aerospace assessments are made at the integrated force, weapon system, subsystem, and technology levels with each of the various divisions within FTD responsible for processing specific categories of intelligence data. The integrated product thus generated by FTD is used for continuing analytical activities within FTD and by external organizations.

2.2 Evolution of the User Communications Concept

2.2.1 STIS Development. One of the automated support aids for FTD analysts is the Scientific and Technical Information System (STIS), an advanced intelligence information mangement system. The in-house developed progenitor of STIS was the Basic-Level Integrated Analysis System (BIAS). BIAS was implemented on the IBM S/360, Model 65 computer using the IBM file management system, General Information System (GIS). BIAS maintained separate files of information which represented Bodies of Knowledge (BOK) that formed the basis for responding to requests for products and information services. When requests for information involved separate BOK's the separate GIS files were redefined as AOR's (Areas of Responsibility) and retrieval was permitted between AOR's, so long as the information in question was marked public.

The AOR capability and the desire to manipulate technical relationships across the spectrum of scientific and technical data resulted in a second generation BIAS based on relational data management technologies, called STIS.

With the introduction of the UNIVAC 1108 at FTD, STIS was reimplemented on the new system using EXEC8 data file and indexed sequential file formats. The implementation included a FORTRAN interface and an interactive interface called the Interactive Processing Language (IPL). The FORTRAN interface allowed users to access STIS in the FORTRAN language, while the IPL interface provided an on-line English-like interface to the STIS information base.

A current STIS effort for optimization and development of the system is aimed at stabilizing the existing STIS and improving performance and functional capability for the automated production environment at FTD.

2.2.2 <u>STIS Application Projects</u>. The relational data structures and information management capabilities of STIS provide special facilities for the correlation, storage, and retrieval of fragmentary intelligence information. As a result, STIS is currently being used by the Intelligence Production System (IPS) and the Integrated Event Analysis System (IEAS). The use of STIS for information management is also being considered for use by the Electronic Warfare (EW) and Command, Control, and Communications efforts at FTD.

2.2.3 <u>User Communications for STIS</u>. In addition to these essentially batch mode applications involving special user interfaces to STIS, an on-line STIS user interface called Interactive Processing Language (IPL) was developed by FTD. Although IPL provided useful on-line access to STIS data, it had a number of limitations, as described in Section 2.4.1;

there was a need to expand the concept of the user interface to provide a capability that would be more powerful, convenient, and flexible. The functional requirements for the proposed STIS User Communications Interface were specified in the course of an initial study contract.[1] A concept was developed that would provide FTD analysts with an effective facility for accessing and manipulating their scientific and technical data.

2.2.4 <u>User Communications for Analytical Tasks</u>. The User Communications study effort was followed by the first stage of a developmental program. During this contract, considerable planning was carried out to provide a language capability that would meet a user's needs. An informal survey of user needs was performed in the early stages of the User Communications contract effort described in this report. This survey was aimed at establishing a set of general requirements associated with interactive use of the STIS information management system, since, at that time, STIS was the focus of the User Communications effort. What emerged from this survey was a coherent view of generic analystical and information processing operations performed by FTD analysts, which could be enhanced by an interactive user interface to automated support facilities including -- but not limited to -- STIS.[1]

The survey involved discussions with FTD analysts from the following groups:

- ETTC
- ETER
- ENDA
- PDS
- XRX
- XRQ

In spite of the different areas of technology in which these analysts operated, there were some striking similarities in the manner in which the analysts performed their tasks. All tasks involved selection or retrieval of data, analysis and computation based on these data, and generation of reports presenting analytical results. Thus, based on what the survey revealed about the analysts' tasks and their work habits, several inadequacies were observed in the User Communications concept as originally stated.

(1) Many analytical tasks involve a mixture of STIS data base access and manipulation activities with other existing intelligence programs.

(2) Analysts may need access to data files not included in STIS.

(3) Producing reports is the end result of the analysis effort. The report generation facilities provided by the User Communications language should not be limited to the data and computations relating to STIS. Rather the analyst should be able to utilize the User Communications system to construct a single routine that

would automatically generate his report independently of where the required data were located or of what programs performed the computation.

The commands and facilities of the User Communications language were thus designed to provide functions for carrying out these analytical and information processing activities performed by analysts. The groups of commands corresponding to the analyst activities (e.g., data selection and manipulation, graphics, text editing and report generation, mathematical, logical and relational) are described in References 1 and 2, together with a suggested implementation using function keys. These groups of command and the proposed function key oriented language approach were then discussed with a smaller subset of the users originally surveyed, who were encouraged to suggest additions of lower level functions they would consider useful in their work.

The design of the User Communications language has thus progressed from the outset with user involvement, and it is critical that such involvement be continued. The breadboard implementation and test stage of user communications development will include user participation as a key feature in the program.

2.2.5 User Communications and the FTD FY83 ADP Concept. In an effort to upgrade the automated support to FTD internal and external users, FTD is considering a concept which calls for the merging of computational resources into a unified facility providing distributed cessing and

versatile communications access. Communications links to FTD computers would be provided not only internally, but also to intelligence and military data networks outside of FTD.

A change of major proportions in the FTD automated support facilities has implications for all existing developmental efforts, and the User Communications program is particularly affected. According to one preliminary version of the concept, some of the considerations which would impact the User Communications capability are the following:

(1) STIS will become potentially available to a wider community of (authorized) users.

(2) Considerable amounts of additional intelligence materials would be potentially available to the analyst and on-line access to these materials could be provided.

(3) Finished intelligence could be available on-line to those who have a need for it. In the FTD environment finished intelligence from one analyst group becomes source information for another.

(4) Analysts could be included in a message dissemination system; User Communications could provide a dissemination mechanism.

(5) User communications could include a capability for specifying collection requirements, based on an analyst's findings during an interaction with the STIS information base.

2.2.6 User Communications -- Revised Concept. In light of the variety of on-line capabilities needed by intelligence analysts and in view of the expanded automated support indicated under preliminary versions of the ADP FY83 plan, the original concept of User Communications as strictly a STIS interface is no longer a viable alternative. More specifically, the FTD User Communications interface could provide a centralized facility for accessing data bases (e.g., STIS) and data files, executing intelligence programs on various processors and generating reports both of preliminary and finished intelligence. In addition, provision could be made to exploit new information from other sources, including document files, messages, requests, etc. User Communications should be fully compatible with and responsive to the environment in which it will be used.

2.3 The FTD Production Problem and the Role of ADP Support

S&T intelligence users comprise a large and diverse market; response to their needs depends on the type and timeliness of the requirements. For decision making, each category of customer requires different degrees of detail and presentation of the "technological" threat. Requirements for advanced approaches in intelligence data processing are constantly increasing. As new sensors are developed, the volume of data to be processed grows vastly. Improved analysis methods require integrated data bases online to the analysts. And, the need to specify changing intelligence collection requirements mandates even more timely reaction to, and analysis of, new data.

In response to the ever increasing demands for more production, FTD is considering a long range plan to upgrade automated support for its intelligence processing operations. The expectation is that individual analysts can perform their tasks better and more quickly, given sufficient assistance from computer systems.

2.3.1 <u>A Two-Fold Problem</u>. The introduction of automation in support of intelligence production can be seen as a two-fold problem, where one aspect involves the acquisition and availability of automated support, while the second aspect involves providing this support in a form such that the analysts it is being designed for will be motivated to use it. The FY83 ADP concept presents a solution to the first problem; however, approaches such as the one we have adopted for the User Communications development are required to solve the second one. Automated data processing history abounds in examples of systems that were developed and little used, because they were designed from the standpoint of the computer system rather than the standpoint of the user.

2.3.1 <u>A Three-Pronged Strategy</u>. In Section 3 we will detail our approach to the problem of providing a User Communications interface that is responsible to FTD needs. Briefly stated, our strategy consists of three points of attack:

(1) develop a breadboard system which is truly user-oriented;

(2) conduct a well planned user training program with pilot groups of user-analysts gaining experience with the system in the FTD environment and providing data for evaluation;

(3) optimize the user interface based upon our experience with the system and the users, and upon their reactions to the system.

By this strategy we expect to arrive at a User Communications system having features such that analysts will find it highly convenient to perform their tasks, automated support will be exploited to the degree possible, and there will be an accompanying increase in analyst productivity.

2.3.3 <u>Automated Intelligence Support at FTD</u>. An important consideration in the development of the User Communications interface is the environment in which it will operate. Current FTD computer installations and program are reviewed below, and some of the proposed plans for the future are discussed.

2.3.3.1 <u>Existing Hardware Support</u>. A number of computer systems already exist at FTD. AD administers a large "general purpose" facility based on a Sperry Univac 1110 mainframe. Some of the other directorates have dedicated equipment for specialized types of processing; specific instances are:

- sensor data processing;
- automated Russian to English translation;
- document retrieval;
- word processing;
- document production;

At present these operations might be characterized as diversified and separate; there are practically no inter-connections between them. In some instances data is transferred from one system to another by the "human link", that is, by hand-carrying a magnetic tape. For example, results obtained on a sensor data processing machine may be taken to the Univac 1110 for further processing or use by other programs.

2.3.3.2 Programs for Scientific and Technical Intelligence Processing. The Scientific and Technical Intelligence System (STIS), together with its information base, resides on the Univac 1110 processor. STIS development was described in Section 2.2.1. To the present, STIS has not been utilized to the fullest possible extent; the principal reasons are:

(1) the lack of a really convenient capability to access STIS on-line;

(2) the contention among users for Univac 1110 resources resulting in a heavily loaded system;

(3) the developmental nature of STIS.

The first problem is being addressed by the User Communications development program, while the STIS Optimization and Development effort is resulting in a more stable system that provides increased capabilities. The second problem can be solved only by providing for STIS a processor with additional resources. STIS supports applications programs, as summarized in 2.2.2.

2.3.3.3 Plans for Upgrading FTD Hardware. Confronted by increasing processing requirements and growing needs for finished intelligence, FTD is considering plans for upgrading its automated support facilities. An integrated concept is being evolved in which development would begin in FY 1980 and would be completed in approximately FY 1983. The automated Data Processing (ADP) concept would enhance FTD capabilities in a significant number of functions by applying state-of-the-art techniques at all levels of computational support. Some of the major emphases being considered in preliminary versions of the ADP concept are described here.

2.3.3.1 FTD Facility Integration. Nearly all of the computing hardware now at FTD would be integrated in a network of processors. This would provide direct data links from one computer to another by means of automatic low-level network protocols. Some of the benefits to be realized at FTD by this integration are:

- distributed processing, e.g., sensor data analysis;
- access to multiple data files and data bases;
- on-line products, reducing publication requirements.

2.3.3.3.2 Terminal System With Communications Interfaces. Based on AN/GYQ-21(V) communications hardware links (DEC PDP-11 computers), an extensive terminal system would provide access to the network of processors described in 2.3.3.1.

2.3.3.3.3 Communications Links With External Communications System. Interconnecting with existing and developmental military and intelligence communications systems would provide immediate access to

a wider range of intelligence sources, a direct link to consumers of FTD products, and a possible network of message systems.

2.3.3.3.4 Replacement of the IBM 360/65 CIRC Computer. The new computer, expected to be connected to the network of processors, would provide computational resources for both CIRC and the FTD S&T data management system, i.e., STIS.

2.3.3.3.5 Addition of PDP-11/70 Processors. Distributed processing would be provided where increased local compute power is required.

2.3.3.3.6 Increased Reliability through the Concepts of Modularity and Backup Systems. Greater fault tolerance would be assured by maintaining identical or similar hardware in many parts of the computational environment. If one system should fail, the availability of a similar one could take over its functions and thus prevent a complete loss of capability.

2.4 Development of a User Communications Interface

Concern for the user and his tasks is at the heart of our design philosophy. By means of a convenient and powerful online language supported by a hospitable interactive environment, the user interface will provide the capabilities required by analysts. A scenario illustrating User Communications interaction in a problem solving task is presented in Section 3. Some of the advantages inherent in our approach are described.

2.4.1 Essential Design Concepts.

2.4.1.1 The User Language. IPL was mentioned earlier as being the currently available capability for providing on-line access to the STIS data base. While it has served a very useful purpose, IPL has certain deficiencies that are remedied in our user language concept.

(1) An "English-like" form. While this is an advantage from the viewpoint of ease of learning, it is also a disadvantage. Complete query forms must be typed in; on a terminal that can not be backspaced, such as the entire line must be re-typed when a typing error occurs. For a non-typist, this often leads to considerable frustration. Moreover, the form of the language is not sufficiently "English-like" in many cases and a user can become confused -- e.g., "What have member of Georgian SSR City having AOR=3?"

(2) Predefined capabilities. IPL is a data base query language in the traditional sense. It provides for operations relating to data base query and update, and does not directly provide for functional operations on retrieved data. (Many on-line query languages support arithmetic computations, sorts, data conversions, and accumulation on data fields and variables: e.g., RAMIS).

(3) Separation of queries. No mechanism is available for operating on data by means of a grouping of related queries. Each retrieval type IPL statement is an isolated independent statement. The system does not provide storage for data extracted in a preceding query to be used in a succeeding one. The only

connection possible between IPL queries is user memory; he remembers the results from one query and applies them in his formulation of the succeeding one. Some query languages have mechanisms to get around this limitation by providing for grouping of language statements, as well as for user-defined variables containing intermediate results.

While the language for User Communications must support STIS access and update, it should provide other capabilities as well. Analysts who require access to the Scientific and Technical data base often need to use this data in the context of a larger task. Data values need to be compared, combined, plotted, or operated on by a range of functions. A substantial amount of an analyst's time is dedicated to writing technical reports. The user interface should provide facilities for this, since in many cases values could be filled in directly from the data base or from routines provided by or constructed from user language commands.

Given that analyst activities are task-oriented, the ideal user language should match those tasks. Our design is based on the concept of language commands in the form of user functions. A problem solving activity is a form of goal oriented behavior, in which a major task is composed of subtasks. A subtask maybe thought of as a sequence of related functions. By designing a language having an extensive repertoire of functions, we are providing the analyst with the tools he needs to perform (primarily on-line) most tasks requiring scientific and technical information stored

in the data base. From a single user interface he will be able to examine (and if necessary, change) his data, perform necessary calculations, and construct a report showing his results.

2.4.1.2 Interactive Environment. While the concept of on-line computing is not new, developers of computer systems have only recently begun to concern themselves with the usability of their systems. Hardware and software resources are very expensive. Automated support can be most cost effective when it allows human users to produce at or near their full capacity. Optimizing man-machine interaction requires careful consideration of human working habits, capacities, memory, physical characteristics, sense data acquisition, etc. These are factors which we may refer to as Human Information Processing (HIP) considerations.

In the design of a user communications interface there are several HIP factors that need to be considered:

- The ease of learning the user language;
- The ease of using the system;
- The ease of entering commands;
- The amount of data that should be presented;
- The form of data display that promotes rapid absorption;
- The form of interaction with the data;
- The methods of recovery from error situations;
- The kinds of messages and prompts the system gives.

A well-conceived user interface should maximize the amount of useful work possible while minimizing user frustration. Careful selection of peripheral hardware devices can significantly increase the two-way transfer of information between a user and his system.

The OSI-PRC concept of the User Communications interface includes a carefully selected "user station" configuration. The functionally oriented character of the user language mentioned above is paired with a functional keyboard; with a single keystroke the analyst will invoke a language function that is an element of his task. The peripherals, including a text CRT and keyboard, a graphics CRT, a graphics data tablet and pencil, and a flexible (floppy) disk drive will be clustered about him in a convenient manner. Hard copy of the displays will also be available when desired.

The user language translation software will employ dynamic scanning of user input; both the text and graphics displays will be active in providing syntactic prompting, instructions for error recovery, explanation of commands, helps and tutorials for use of the system and access of the data base. User data structures, such as arrays, that must be "linearized" for input from the text terminal will be dynamically displayed in their more intuitive form on the graphics CRT.

Users will be able to progress very rapidly through STIS data base structures by interacting (through the data tablet) with the graphics displays. A rich repertoire of graphics menus will enhance the power of the user language. Direct interaction with the displays will provide

a dimension of information flow that is not possible with strictly text input-output. The primary interactions will consist of function key strokes and data tablet point picking. Typewriter keyboard input will be necessary only for entering character string and numerical data.

2.4.1.3 Interface Capabilities. The user station concept described above will provide the analyst with all the automated resources he needs for his scientific and technical intelligence processing tasks. The interface will provide on-line access to STIS, access to existing intelligence and plotting programs, access to non-STIS files such as sensor data, etc.

Given these capabilities, the user can be expected to interact at his station in various modes.

System Mode. The analyst is communicating with the User Communications system itself or setting up communications with another processor on the network. He receives messages, sends mail to other users, or asks for guidance in performing an operation.

Explore Mode. The analyst is interacting with STIS to discover the nature and extent of available data for a study. He looks at his text files from previous sessions to refresh his memory on the progress of the task. He looks at various values of data base attributes while deciding which ones to use. This mode is for memory-refreshing, thinking, and planning for his task.

Work Mode. The analyst begins to perform the sub-tasks that are necessary to his study. He defines new functions in the user language and modifies old ones. He retrieves data, stores new values, and creates new relationships between STIS nodes. He calls for batch execution of programs on the large mainframe. He does computations locally using the user language functions.

Compose Mode. The analyst performs a special subtask -- the preparation of a report on his study. He investigates the report formats he and others have already defined in the user language. He modifies one of these, or finds them inadequate and builds a new one. He creates a user function to insert the text, the calculations, and the graphics materials at the appropriate places on the pages of the report.

Generate Mode. The analyst checks that all of the text material and data are correct. Then he presses the function key to begin generation of the report. He looks at one or two pages on the CRT display and decides everything is as planned. He then presses the appropriate function keys to route the output to the printer/plotter for hard copy. He also saves a copy in a file for use on-line by his "consumers".

3.0 THE USER'S VIEWPOINT

In this section User Communications will be looked at in terms of the impressions, reactions, and responses a new user might have in an initial encounter with the system. The novice (whether he is already fluent in "computerese" or not) will be conducted personally and progressively on a guided tour of User Communications capabilities, from an introduction to the supporting hardware to an instructive first online session that will give him a feeling for the ways he can expect the system to work with him in the performance of his assigned tasks. What is presented here is intended to convey a concept rather than specify the exact hardware configuration. The user will become acquainted with the User Communications environment, after which he will be led from the simple to the more complex capabilities of the user language.

3.1 The User Interface Environment

3.1.1 Getting Acquainted with a User Station. As the analyst approaches the area dedicated to User Communications work stations, some observations of a general nature are pertinent. Although the system designers have no control over the physical environment, it should ideally be a pleasant place to work; the lighting should be adequate and not glaring, the temperature should be regulated, and noise levels should be comfortably low.

We indicate to the analyst a particular location where he is to have his first experience with a user station. Entering a cluster of equipment, he takes a seat in the user's chair; he finds it reasonably comfortable, and

the swivel base allows him freedom of movement and access to the various items of equipment in the cluster. As shown in Figure 3-1, computer peripheral equipment is organized around a U-shaped table, with shelves and drawers below for supplies and accessories.

Suspended above the table top at eye level are two CRT display devices. On the screen to his left the analyst notices the words, "WELCOME TO STIS USER COMMUNICATIONS". The other screen contains a drawn figure, which he recognizes as an outline sketch of the user station at which he is now seated. Below the displays at the center of the table he observes a keyboard panel with a cable connected to it. To the left he sees what resembles a typewriter keyboard with more keys than normal, and to the right a large flat rectangular device, also with a connecting cable.

Our new user looks beyond the table at which he is seated and notices some other device between his work station and the next one. He then turns his chair a full ninety degrees to the left and sees on the table something that resembles a metal box, with two slots in the front.

3.1.2 <u>Finding Out What the Equipment Does</u>. When we feel that the analyst has had time to digest his initial impressions of the user station, we inform him that the station has its own familiarization exercises and suggest that he start by depressing a prominent button marked "HELLO" on the keyboard panel in the center of the table. When he depresses the key, the display on his left is erased and the word HELLO appears at the top of the screen. Immediately, underneath he reads, "Do you have a log on

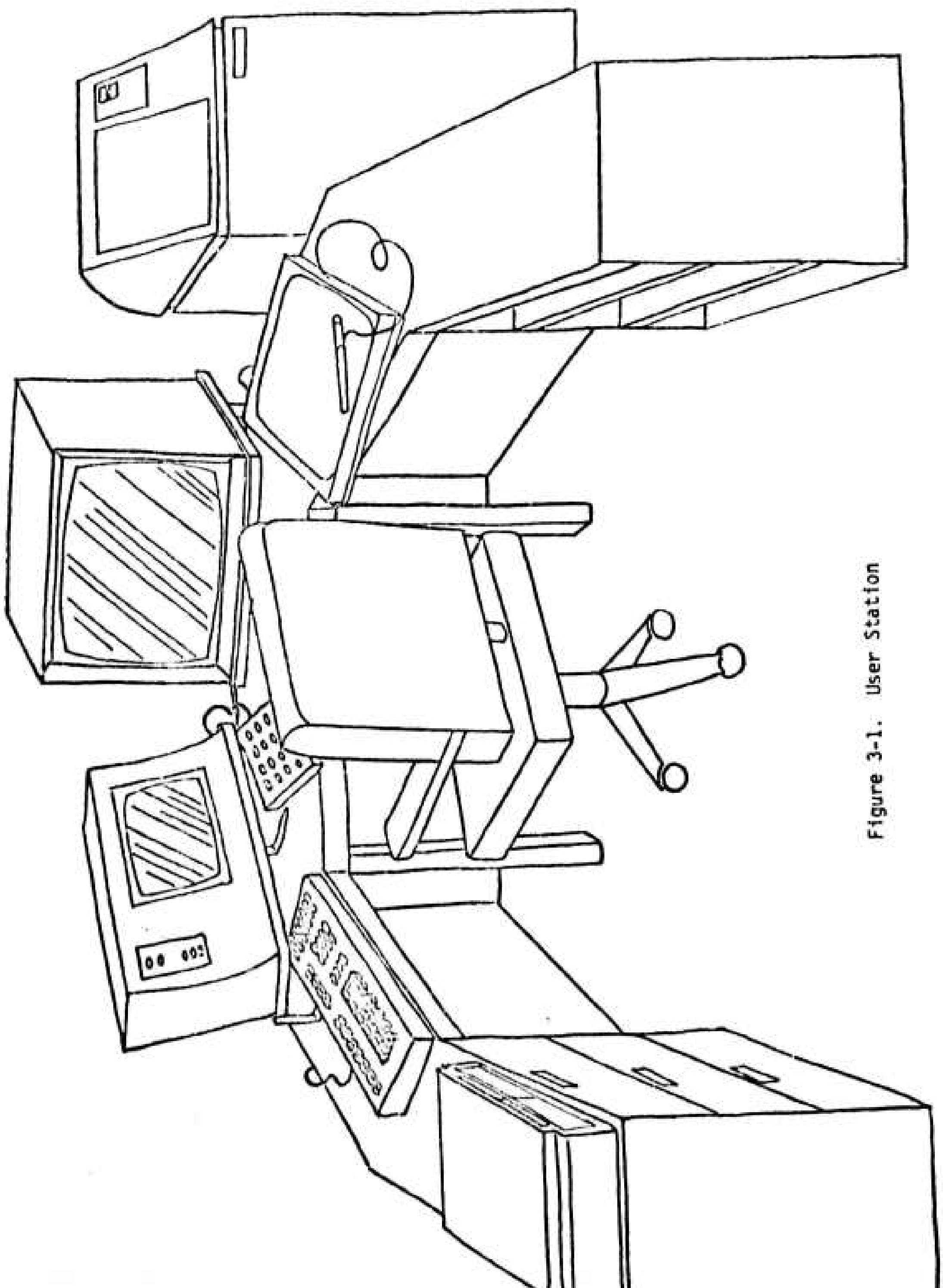


Figure 3-1. User Station

identifier and password? Answer by typing Y for yes or N for no on the typewriter keyboard on your left." He types an N, and the display on his left adds the following message:

"Log on identifiers and passwords may be obtained from the STIS Data Base Administrator, with proper authorization from your supervisor. Since you are apparently new to User Communications, let us explain what the user station equipment does and how you operate it. You will notice on the same keyboard panel you used before 2 other keys, one marked STOP and another marked CONTINUE. Pressing STOP will terminate the session; CONTINUE will cause the next display or other instruction to be given. This instruction is now under your control".

When nothing more happens after a few seconds, the analyst depresses the CONTINUE key on the panel. The system responds by informing him that the picture on the right hand display will form the basis of the next few lessons. He looks at the right hand display and observes that the drawing has changed; the image corresponding to the device under and to the right of this display has been over-written with the words "DATA TABLET" in blinking letters. The other display tells him that the labelled device is to be used by him to reference parts of the picture display. Following the instructions, he finds a pencil-like object attached to the rectangular data tablet by means of a small wire. He picks up the "pencil", and as he moves the pointed end in close proximity to the top of the flat surface, he notices a faint line describing the movement being traced on the righthand display. He touches the data tablet surface at a point where a dot was shown inside the representation of the righthand display, and that part of

the picture receives a blinking label, "GRAPHICS DISPLAY". The label on the picture of the data tablet now stops blinking.

The lefthand display tells him that the graphics display and data tablet form one of the three principal forms of interaction with the system. The graphics medium, emphasizing the spatial and dimensional aspects of data, has several important uses, and a significant number of tasks can be done using only the data tablet and graphics display. He is told what a graphics "menu" is and how he may select menu items he sees on the graphics screen, using the data tablet and pencil.

Having a "handle" on graphics, the analyst next touches the area of the data tablet that shows a dot over the lefthand display device; the blinking label he gets is "TEXT DISPLAY". He is told that the primary keyboard for this display is below and to the left of the display itself. The keyboard is in fact like a super typewriter, containing numbers, letters, punctuation, and "special characters". All except the latter are "echoed" on the screen as the keys are depressed. He observes that, unlike a typewriter, this device has a special marker on the screen (called a cursor) that indicates where the next character will be typed. He discovers that there are special keys for moving the cursor left and right, as well as up and down, and he can even make it blink. He notices that under certain conditions the cursor will erase existing characters as it passes over them, while under other conditions it will not. The instructional information presented to him on the text display states that the text system has a number of capabilities and features available by means of the special characters and special keys

on the keyboard. The menu on the graphics display allows him to select more detailed instructions; he realizes that the text system is more complex than he thought and decides to defer further exploration of the typewriter keyboard for a later session.

Pointing to the area of the data tablet corresponding to the picture of the function keyboard, the analyst selects a description of the third major interactive component in the User Communications system. From the explanatory messages on the text display he learns that this keyboard provides the primary input capability for the language he will use to communicate with the User Communications system. The keys provide the functions he will require in his work. The basic keyboard has keys arranged in rows, with numbers opposite them; there are 20 or 30 keys on the board. The coding capability of the keyboard is greatly expanded by the use of overlays, which are transparent pieces of plastic material with holes for the keys. The overlays fit over the top of the keyboard, as shown in Figure 3-1A, and the label adjacent to each key shows the use (function) of that key. The overlays change the electrical coding of the keys, so that each overlay gives the keys a different meaning. Thus using ten (10) overlays with thirty (30) function keys, it is possible to generate 300 distinct functions. Overlays not in use will be kept in a drawer of the table.

User functions are organized by overlays; this allows a particular kind of task to be performed without changing the overlays. The tutorial explains the various overlays of the function keyboard, listing the kinds of things that can be done with each. One, he is told, activates keys for executing

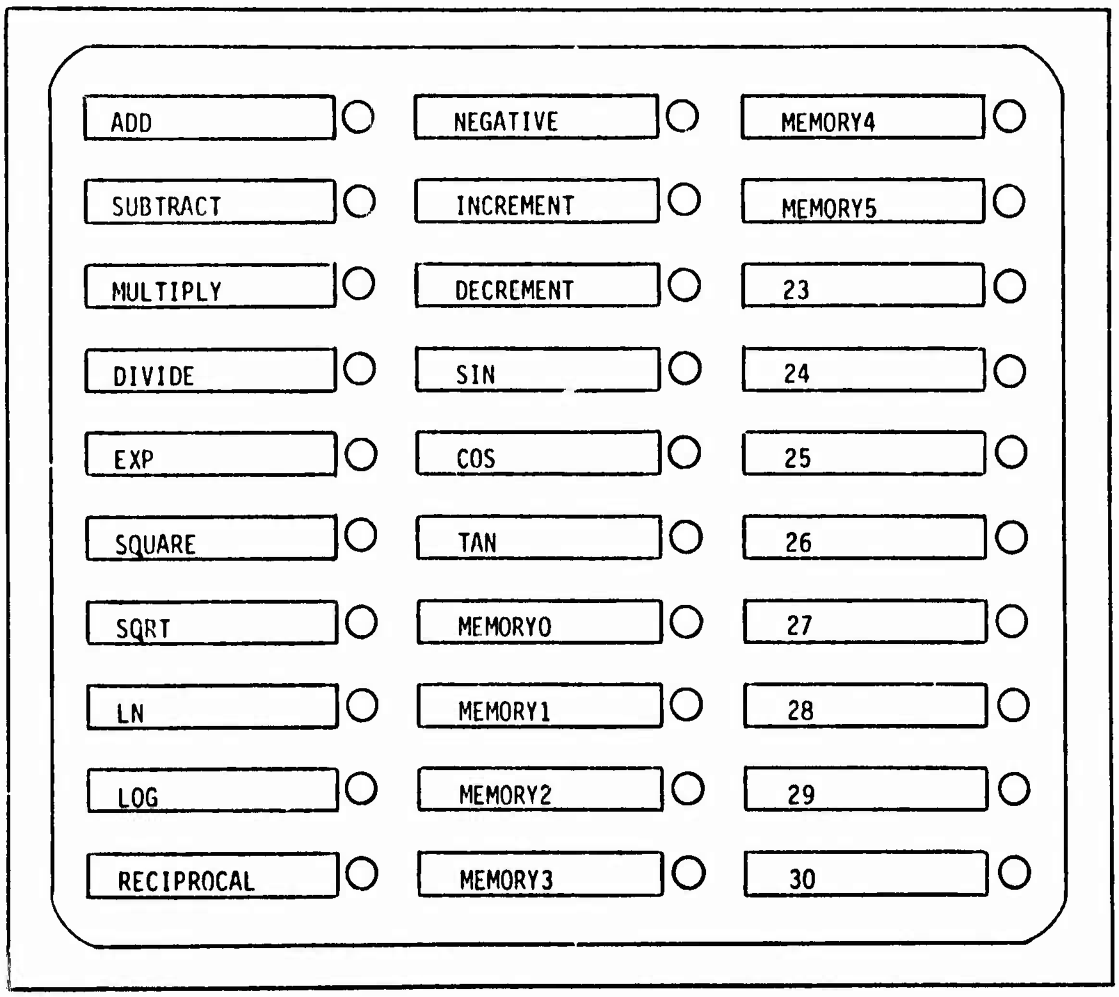


Figure 3-1A. Function Keyboard with Arithmetic and Math Overlay in Place

arithmetic and mathematical functions; another has to do with access to the STIS data base, while a third has the commands necessary for editing text and generating reports. The analyst notices that he may freely intermix keys from the function keyboard with those from the typewriter keyboard and that the display of the function names on the text screen is always surrounded by blanks.

All of the function keys, he discovers, are echoed on the text screen; however, he is informed that the effect of pushing a key will sometimes be evident on the text display and at other times on the graphics display, depending upon the nature of the function. He learns that many of the language functions or commands relating to graphical displays are duplicated in the graphics menus. He thus has the choice of selecting these functions by depressing a key or by pointing to the name of the function on the graphics display with the data pencil; he does not have to change the mode in which he is interacting with the system.

All of the equipment on the table has now been explained except the box directly to his left. He points to the picture of the box on the graphics display and is informed that it is a floppy disk device. He learns that this equipment provides storage for information he may want to keep when he begins to use the system on a regular basis. Following the instructions he pushes open a door behind one of the slots in the front of the device. He removes a flat round di ' that resembles a slightly enlarged 45 rpm record. This disk, he is told, is the storage medium for his data.

Other types of storage media are rigid and larger in diameter; for this reason the one he is holding in his hand is called a "diskette" or a flexible or "floppy" disk.

The diskette device holds two diskettes, he learns, and it can read or write information on one or the other (or both) independently. In this way, information can be copied from one diskette to another. He is told that data is stored on the diskette in named units called files; a file can be named by him or by the system--the option is his. After a typical session at the user station, he will be able to write data he wants to save in files on his own personal diskettes and take them to his office for use in a later session.

The analyst now notices that the only device that is still a stranger to him is the one between his station and the neighboring one. He touches the appropriate area of his data tablet and discovers that it is called a printer/plotter. He wonders out loud whether this could be the missing link between what he sees on the two displays and something more permanent that he can examine at his leisure. The tutorial immediately assures that this is the case; among the several uses of this device, it makes available "hard" (i.e., paper) copy of the information on the text and graphics screens.

Our friend learns that the price to be paid for a single machine providing both textual and graphical material is speed. Fortunately, he does not have to wait for a page to be completed, however; the printing is scheduled and controlled by the central minicomputer as an activity that

is independent of his work station. Once he gives the appropriate command for printing, he can forget it and go on to other tasks. The printer/plotter is a resource that is shared among the user stations. When an analyst wants to see his hard copy, he will go to the device and remove those pages that he caused to be generated. Besides screen images, he can print his files and other kinds of direct output from his station. By now the analyst feels that he has a reasonable understanding of how the work station equipment functions. He has spent some time getting familiar with each device. He realizes that there are many facts and details he still doesn't know, but he has reached a kind of mental saturation with this particular acitivity. He is confident now that he can return at a later time and teach himself, with the help of the system, to become proficient and even skillful in operating the user station.

3.1.3 Putting User Communications into Perspective. As we walk with the analyst back to his office, we explain some of the important User Communications concepts. He has heard a great deal about the STIS information base, but his understanding relates more to its contents than to general knowledge of the system. We explain that STIS is located on a large computer, which is linked to the user station minicomputer. When a user at any of these stations pushes a key that involves data base access, the minicomputer formulates a request which is sent to STIS on the large data base computer. STIS executes the action defined by the request and sends back data or some other form of acknowledgement. The minicomputer routes this information back to the appropriate user station, where it is translated into messages or drawings or both.

Having seen the minicomputer on his way to the user station, the analyst does not find it difficult to understand the operation of STIS data base queries. What he did not realize is that User Communications provides many other capabilities as well; it is a general facility for on-line access to programs and data. For example, commands are provided to set up and run sensor processing data on one of the large computers, if a user has a requirement to do this. He may also retrieve data from files generated by these programs on the large system. In this application the user station is functioning as an RJE (remote job entry) terminal. The minicomputer is one of the nodes in a network of computers at FTD, and the user stations provide communications access to this node.

The analyst quickly perceives the advantages of network participation; nearly all the information resources he might require for his job are readily available from his work station. He will be able to retrieve documents that relate to his studies and insert relevant parts of them in his reports. He will be able to generate messages and on-line reports for those who need to use the results of his work. He will have convenient facilities for presenting and perhaps even routing on-line requirements for additional intelligence collection. Whereas our user communications initiate had previously been mildly amazed with the versatility he had observed in the user station equipment, he now begins to experience the potential of the new horizons offered by User Communications for carrying on his S&T analytical tasks.

3.2 The User Language

One of the best means of explaining the user language functions is by examples. Two related scenarios are presented here, one for solving a specific problem and the second for generating a report relating to the problem solution. These scenarios illustrate a small representative subset of user language commands, but more importantly show their operation in the User Communications environment.

3.2.1 A Problem Solving Scenario. To show the utilization of the User Communication Interface in FTD intelligence production, a trivial example of the identification of an ICBM in a given firing event is described. The intention of the example is to demonstrate a typical interaction session rather than to present a truly valid algorithm for the identification of ICBMs.

The hypothetical problem of the missile event can be described as:

- identify the ICBM as a known configuration or
- identify the ICBM as a new system
- produce a report on the parameters of the event.

The data available about the event include:

- Launch site: Aktyubinsk, USSR missile launch facility
- RV impact: 6° No. Lat., 152° West-Log (Near Christmas Island)
- 3RD stage ignition; altitude of 90.6 KM over ULAN BATOR, Mongolia
- 3RD stage ignition: 1327 hours GMT

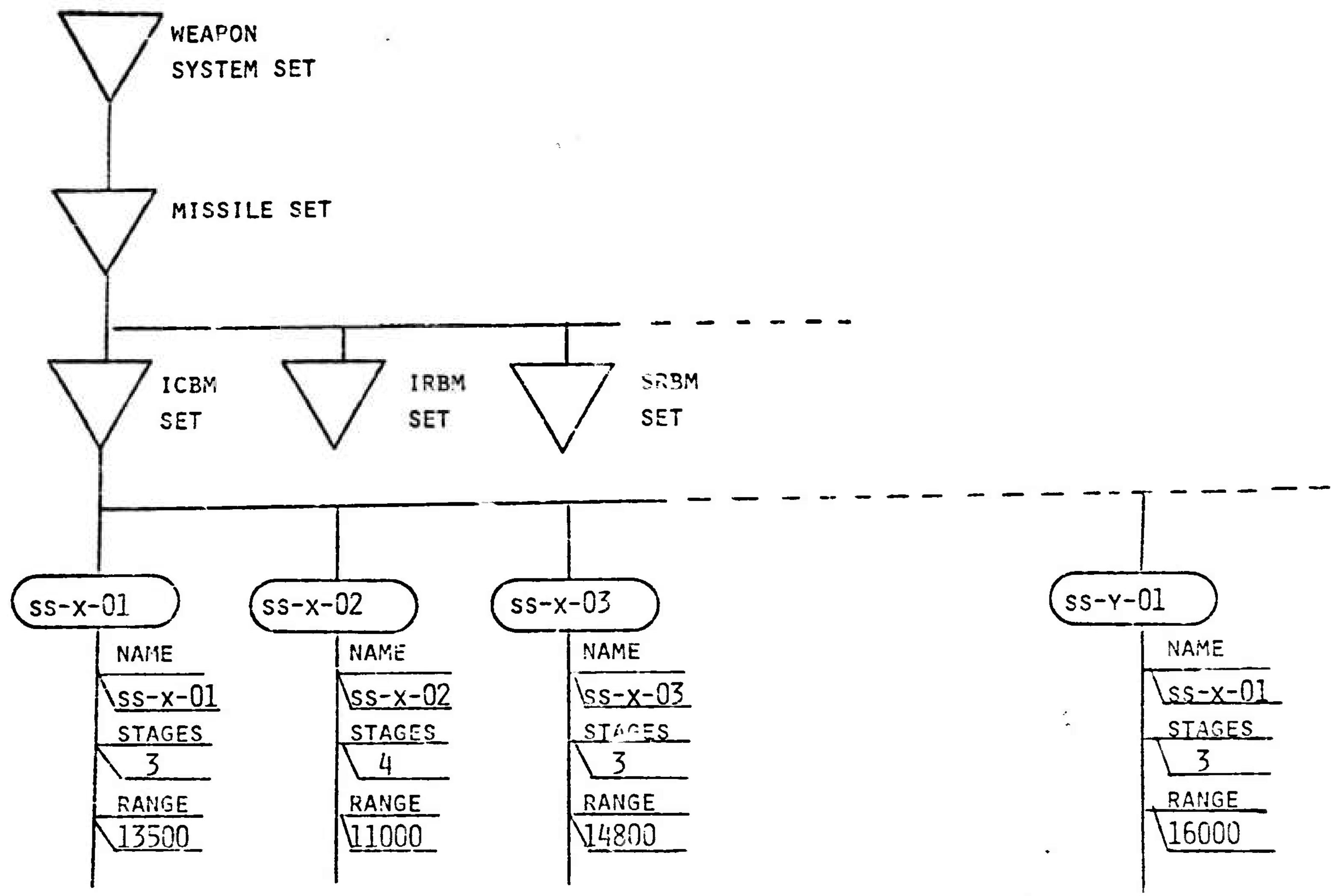


FIGURE 3-2. STIS ICBM DATA BASE STRUCTURE (ABBREVIATED)

FUNCTION KEY PRESSES/ MENU SELECTIONS	VARIABLES TYPED IN	DESCRIPTION OF COMMAND FUNCTION
DEFINE_LOOP_FUNCTION	"LOCATE CANDIDATE ICBMS"	USER DEFINED FUNCTION TO LOCATE CANDIDATE ICBM SYSTEMS IN THE STIS INFORMATION STORE
SET_CP	"ICBM"	
RETRIEVE	"STAGES"	RETRIEVE NUMBER OF STAGES OF CURRENT ENTITY
IF	STAGES	
EQUAL	3	IF NUMBER OF STAGES=3
CONTINUE		CONTINUE
HOPDOWN		GET NEXT ICBM
RETRIEVE	"RANGE"	RETRIEVE RANGE OF ICBM
IF	RANGE	IF RANGE EQUAL TO OR GREATER
EQUAL_OR_GREATER	30000	THAN 30000 CONTINUE
CONTINUE		
HOPDOWN		ELSE RETRY FUNCTION
RETRIEVE	"NAME"	RETRIEVE NAME OF ICBM
ADD_LIST		ADD ICBM NAME TO CANDIDATE LIST
END_FUNCTION		END
DEFINE_FUNCTION	"IDENTIFY ICBM"	USER DEFINED FUNCTION TO CREATE AN IDENTIFIED ICBM ENTITY IN THE STIS INFORMATION BASE
CREATE_MEMBER	"ICBM"	CREATE NEW ICBM ENTITY AS A MEMBER OF THE ICBM SET
STORE	"NAME"	STORE ICBM NAME IN STIS
	NAME	IN LOCATION CALLED <u>NAME</u>
END_FUNCTION		END

Figure 3-3. Illustration of Procedure for Creating User-Defined Functions

GRAPHICS CRT DISPLAY

FUNCTION: "LOCATE CANDIDATE ICBMS"

VALUE: "ss-x-05"
"ss-x-06"
"ss-x-09"
"ss-x-11"

ALL_FACTS	ALL_ATTRIBUTES	SET_CP	REVIEW_DEFAULT_FCI
SCALAR_FACTS	SCALAR_ATTRIBUTES	LINK	MODIFY_DEFAULT_FCI
RELATIONAL_FACTS	RELATIONAL_ATTRIBUTES	MORE_DATA	MODIFY_FACT_FCI
			MODIFY_FACT_FCI

TEXTUAL DISPLAY

LOCATE_CANDIDATE_ICBMS

Figure 3-4.. Values Returned by User-Defined Function Shown in Figure 2.2

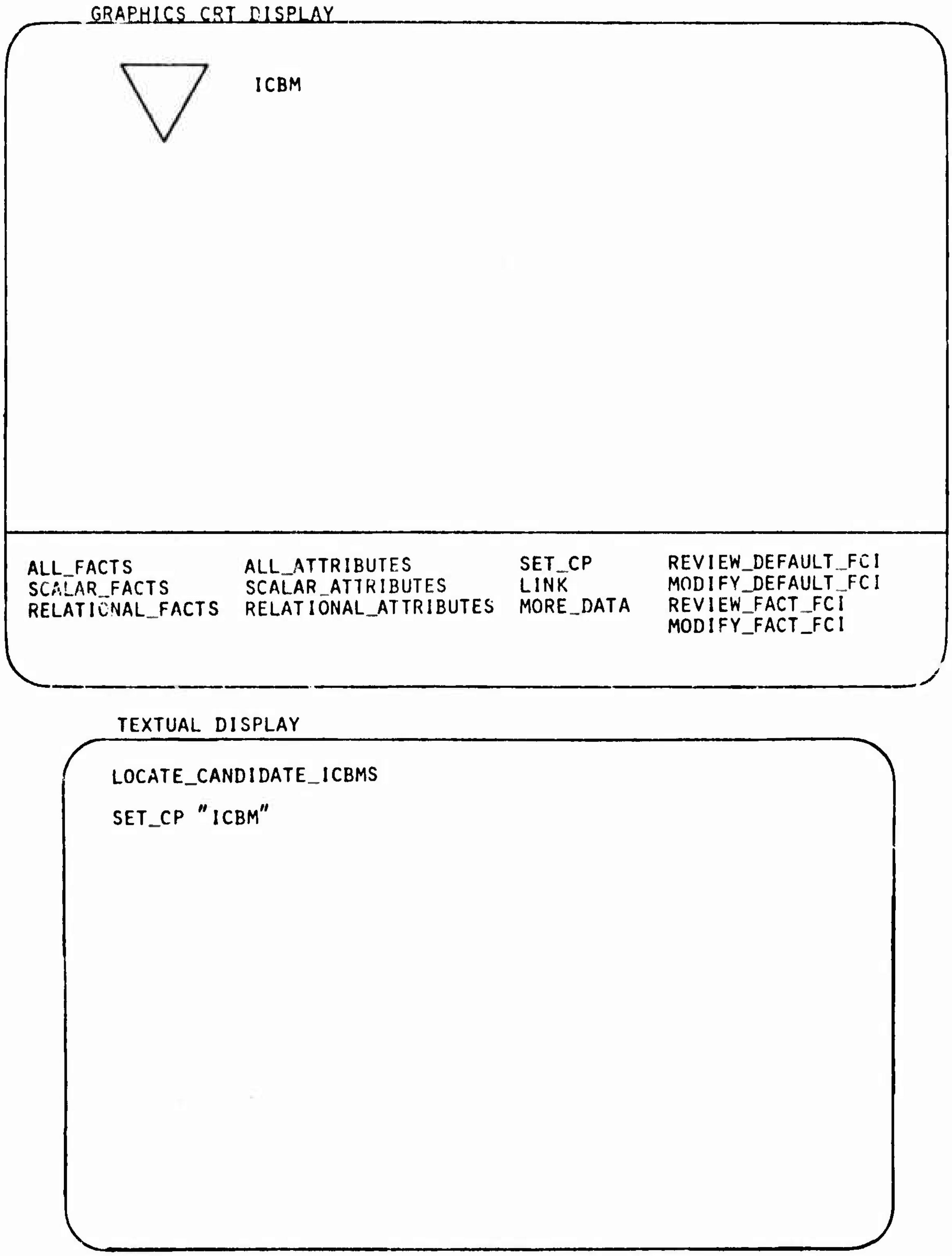
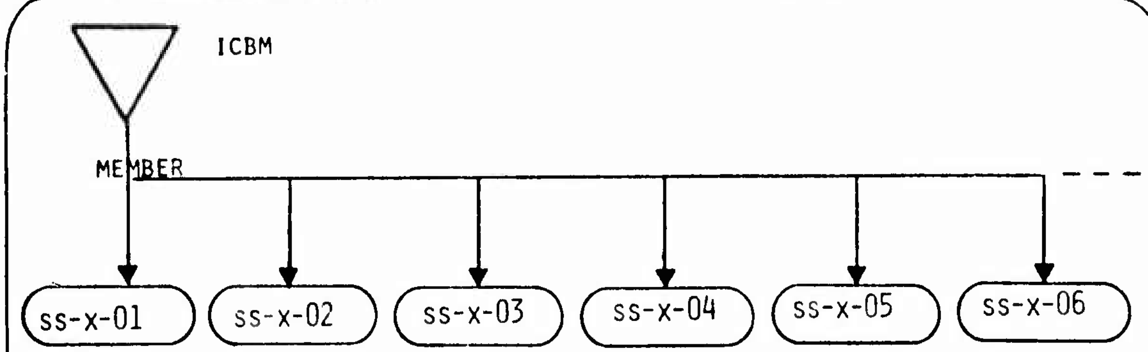


Figure 3-5. Effect of "Set Context Pointer" Command

GRAPHICS CRT DISPLAY



ALL_FACTS	ALL_ATTRIBUTES	SET_CP	REVIEW_DEFAULT_FCI
SCALAR_FACTS	SCALAR_ATTRIBUTES	LINK	MODIFY_DEFAULT_FCI
RELATIONAL_FACTS	RELATIONAL_ATTRIBUTES	MORE_DATA	REVIEW_FACT_FCI
			MODIFY_FACT_FCI

TEXTUAL DISPLAY

```
LOCATE_CANDIDATE_ICBMS
SET_CP "ICBM"
RELATIONAL_FACTS
```

Figure 3-6. Displays After 'Relational Facts' Command is Entered

- 3RD stage burn time: 9.23 minutes
- 3RD stage maximum altitude: 139 KM
- RV impact time: 1405 hours GMT
- Total observed range: 16,575KM.

The strategy of the analyst is to use a predefined User Communications function to locate all ICBMs that have three stages and a range greater than that of the one observed. Using the attribute and fact functions, he examines the candidate systems in detail via the graphic displays and then creates a new specific ICBM, using a predefined function, of the unidentified type in the STIS information base.

The data base structure being examined appears as shown in Figure 3-2. The two predefined functions to be used in the identification process are illustrated in Figure 3-3. In the process of creating these function definitions, the user presses function keys (or makes menu selections with light pen or data tablet) to enter the commands (Column 1) and types in associated variables (Column 2). A brief description of each command function is given in Column 3.

After a greeting display containing an initial option menu, the analyst presses the function key associated with the "LOCATE CANDIDATE ICBMs" function (assuming this has been defined previously). The graphics CRT display returns with the list of candidate ICBMs and the function name. The textual display contains a textual image of the command, as shown in Figure 3-4.

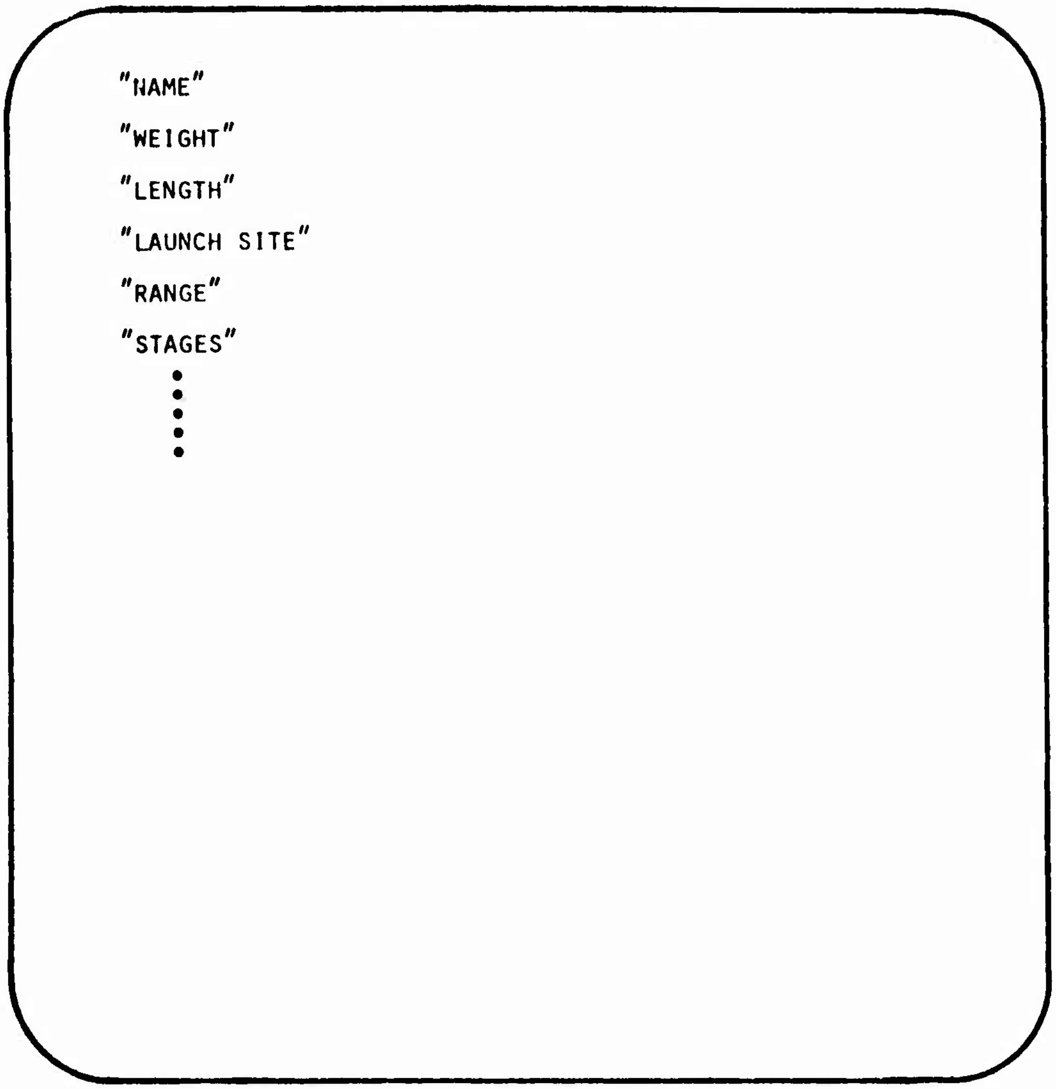


Figure 3-7. Textual Display Showing Attributes List

"NAME" = "SS-X-05"

"WEIGHT" = 2280KGM

"LENGTH" = 206M

"LAUNCH SITE" = "AKTYUBINSK, USSR"

"RANGE" = 31,000KM

"STAGES" = 3

Figure 3-8. Textual Display Showing Values of Attributes

Figure 3-9. Textual Display Showing Multiple Values

FCI FOR WEIGHT = 2280 KG

SIA = FTD

INT = HUMINT

SEN = FTD ANALYST DERIVED

CFL = COMPLETELY RELIABLE

AOR = PDBA

CLS = UNCL

RFS = UNDEFINED

RFN = UNDEFINED

RLS = UNOFFICIAL

SCR = PUB

DOB = 770302

Figure 3-10. Fact Control Information Display

The analyst then touches SET-CP (set context pointer) on the data tablet and enters "ICBM" as the set name on the textual display. The two displays now appear as shown in Figure 3-5.

The analyst now desires nore detail about the members of the ICBM set and touches the RELATIONAL FACTS option on the data tablet of the graphics CRT. The display that results is the data structure subordinate to the ICBM set, which appears as shown in Figure 3-6.

If the analyst now desires the attributes of a particular ICBM, he would touch the ALL-ATTRIBUTES option on the data tablet of the CRT, then the desired node, and the attribute list would be displayed on the textual CRT, as shown in Figure 3-7, leaving the graphics display unchanged.

The analyst may obtain the values of the attributes by selecting the ALL-FACTS option on the data tablet of the CRT. The list of attributes and values would then be displayed on the textual CRT, as shown in Figure 3-8.

The analyst can peruse the data base by various selections of the FACT and ATTRIBUTE commands, with scalar responses appearing on the textual display, while maintaining the integrity of the graphics display. When the ICBM has been identified, the analyst selects the "IDENTIFY ICBM" option key and enters the ICBM name into the textual terminal.

FCI Display. The analyst has the option of reviewing and/or modifying Fact Control Information of either the Default FCI or the FCI of individual facts. A typical textual display containing a list of facts is shown in Figure 3-9.

The WEIGHT attribute has two conflicting values of 2280KG and 3680KG. The rationale for the differences in weight can be found in the FCI fields for these values. The analyst would select the REVIEW-FACT-FCI option, select the desired fact, and the resulting textual display would contain the fact control information, as shown in Figure 3-10.

By selecting the FCI for the next value, the analyst would observe that Confidence Level (CFL) would be UNRELIABLE, which would indicate the reason for the two conflicting values. If the fact is stored under the analyst's AOR (Area of Responsibility), the FCI control information could be changed to reflect the current state of affairs by selecting the MODIFY-FACT-FCI option. This would require the typing of the new fact control values onto the textual display.

In a similar manner the default FCI could also be reviewed and modified, except that default FCI is global and not local to individual facts.

3.2.2 A Report Generation Scenario. Having found a satisfactory solution to his problem, the analyst now proceeds to produce a report detailing his conclusions. It is assumed that the report consists of two parts, a one-page tabular summary of the important information followed by a detailed textual description.

3.2.2.1 The Summary. This part of the report will use one of several existing tabular forms. The summary contains two primary types of information -- background or historical and new information. Information in the table is partly numeric data and partly textual. An example of the background data might be:

```
          UPDATE REPORT ON ICBM SYSTEM - - - - - -

XXXXXXXXX     :  - - - -          XXXXXX     :  - - - -
XXXX          :  - - - -          XXXXXXXXX  :  - - - -
XXXXXXXX      :  - - - -          XXXXXXXX   :  - - - -
XXXXX         :  - - - -          XXXX       :  - - - -
XXXXXX        :  - - - -          XXX        :  - - -
XX            :  - - - -


           XXXXXXXXXXXXXXXXXXXXXXXXXX:  - - - - - - - - - -
           - - - - - - - - - - - - - - - - - - - - - - - -
           - - - - - - - - - - - - - - - - - - - - - - - -
```

DISPLAY ON TEXT CRT

Figure 3-11. Display of Report TABLE17

Date of first observed firing: 8/14/71,

while items of new information could be:

Configuration changes: Stage 3

Effects of Changes: ICBM range extended to 16575 KM

To produce the report summary, the analyst will adopt the following procedures:

(1) Find the pre-stored table which presents the desired information in the correct form;

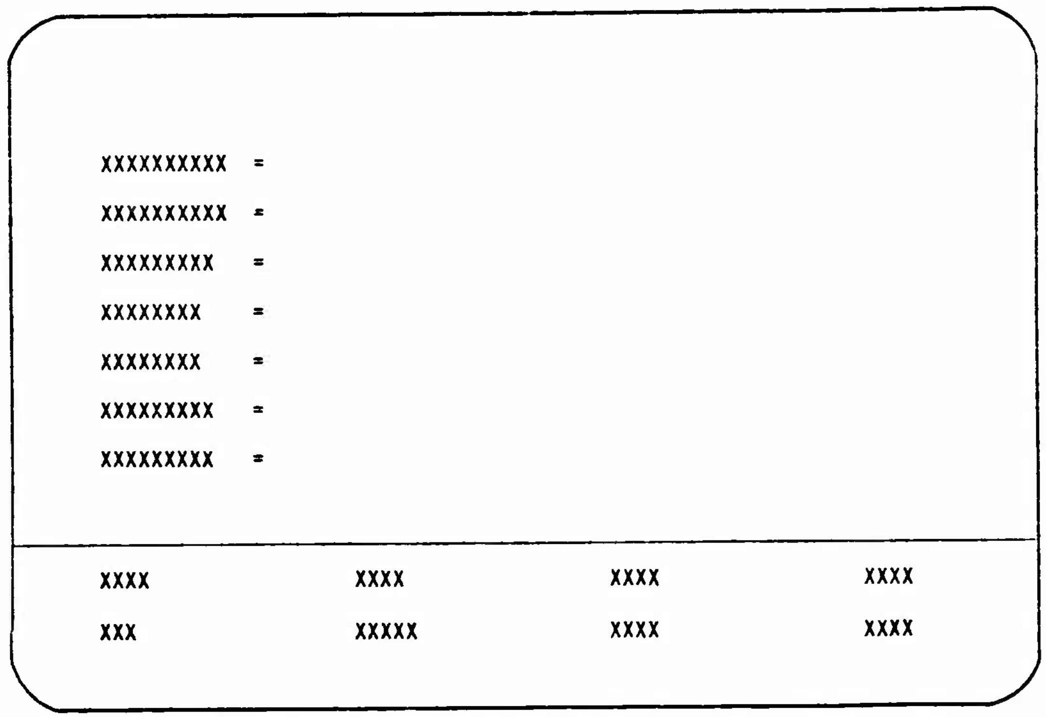
(2) Prepare data in a form that can be used by an existing function that the analyst has written to fill in the data fields in the correct table;

(3) File the completed table and print a copy.

Not remembering which tabular form is the correct one, the analyst looks in a file containing descriptions of the various tables that are available. The command:

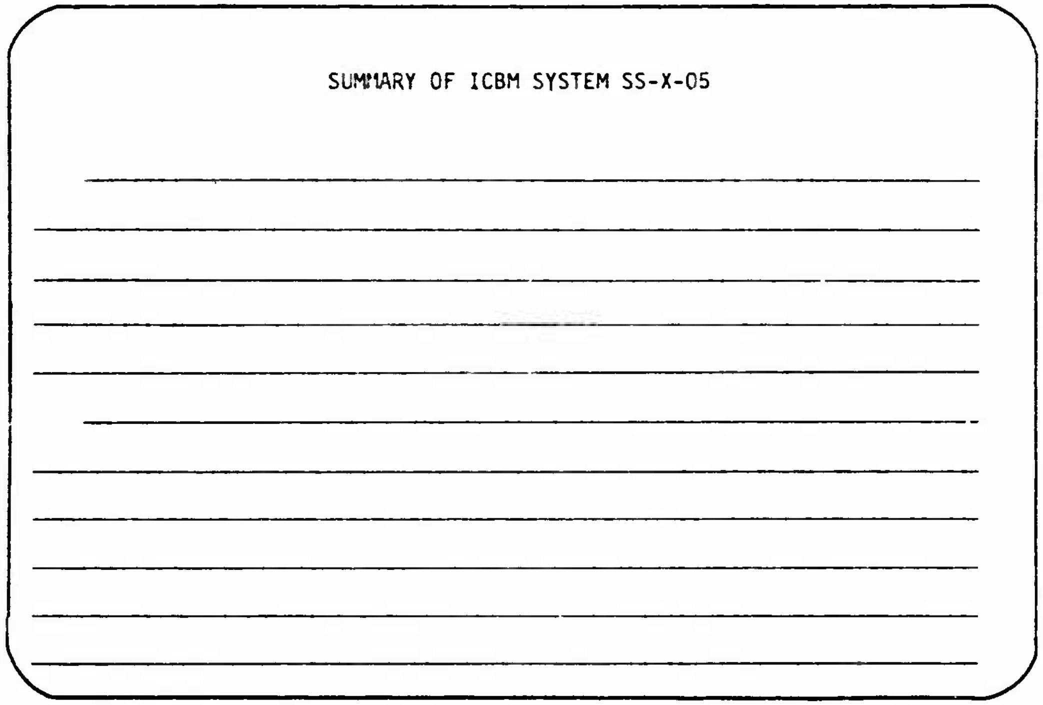
GETFILE REPORT_FRAMES TEXT XEQ

tells the system to route the desired file to the text display (the name of the file is input from the typewriter keyboard; all other parts of the command are on function keys). The analyst scans the descriptions provided, using the scrolling capability of the text terminal. He discovers that the table he wants is identified in the system as TABLE17. He displays the table using the command:



Graphics Display

Figure 3-12. Display of TABLE17-DATA



TEXT DISPLAY

Figure 3-13. Display of STATS_SS-X-05

```
                UPDATE REPORT ON ICBM SYSTEM SS-X-05

XXXXXXXXXX     :  YYY              XXXXXX      :  YYYY

XXXX           :  YYY              XXXXXXXX    :  YYY

XXXXXXX        :  YYYY             XXXXXXXX    :  YY

XXXXX          :  YY               XXXX        :  YYY

XXXXXX         :  YYYYYY           XXX         :  YYYYY

XX             :  YYYY


          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:  YYYYYY

          YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

          YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
```

DISPLAY ON TEXT CRT

Figure 3-14. Display of Completed Report Summary

GETFILE TABLE17 TEXT XEQ.

Figure 3-11 shows the gross characteristics of the table, as presented on the text screen.

The system provides two ways for filling in such a table; using one method the text terminal provides protected text fields within a line, so that required data can be typed into the unprotected parts of the line. The completed screen then becomes a new image, which is saved. The second method consists of building a special data array, with help from the system, and using this array in a function built from user language commands to fill in the table. Since the analyst already has such a function, called FILLTABLE17, he decides to use the second method. He retrieves a file which assists him in filling the auxiliary data array by entering the command,

GETFILE TABLE17_DATA GRAPH XEQ.

His file is presented on the graphics screen, as shown in Figure 3-12. He recalls that he has already accumulated much of the required data for a previous report, so he brings it in to help him fill in the TABLE17_DATA array by entering the command,

GETFILE STATS_SS-X-05 TEXT XEQ.

Using the powerful edit capabilities, he first updates the file, which he now has on the text terminal (see Figure 3-13). He then transfers the

necessary data to the TABLE17_DATA array, which is still displayed on the graphics screen. Desiring to save the file he has just updated, he enters the command:

PUTFILE STATS_SS_X-05 TEXT XEQ.

He is now ready to build the report table summary, and he does this by calling the predefined user function:

FILLTABLE17 STATS_SS-X-05 XEQ.

Figure 3-14 shows the completed table, as presented on the text CRT. The analyst needs to save this table for later use, and he does so, assigning it the name REPORT259 in the command:

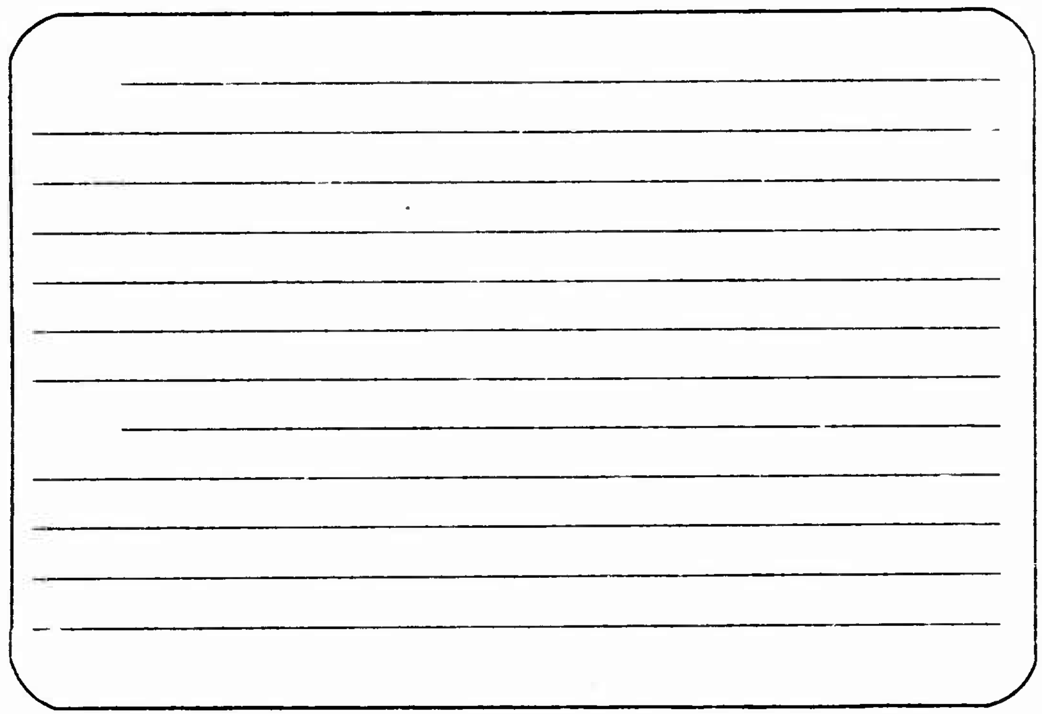
PUTFILE REPORT259 TEXT XEQ.

Also he produces a copy on paper by sending the file to the printer:

PRINTFILE REPORT259 XEQ.

3.2.2.2 The Body of the Report. In the main section of the report, the analyst describes the problem that was given and how he solved it. The report will have several paragraphs, as follows:

- Information about the electronic lock-on and tracking of the ICBM in flight;
- Summary of significant sensor data;
- Summary of the procedure used for identifying and cataloging the new SS-X-05 version;



Text CRT Display

Figure 3-15. Creation of Text Material for Report

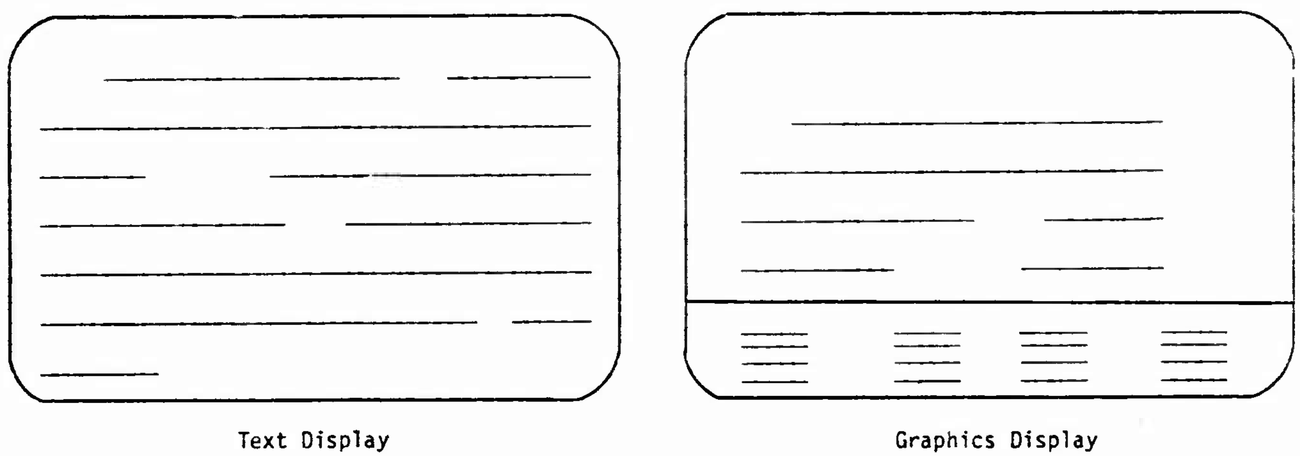


Figure 3-16. Two-Screen Composition

- Conclusions about the SS-X-05 development;
- Indications of uncertainty about aspects of the new SS-X-05 version;
- Plots of significant data;
- Requirements for collection of new intelligence on future firings of this system.

The analyst will approach this report writing task with the following strategy:

(1) He or his secretary will, using the text terminal, generate pieces of textual material in any order that he finds convenient. Each piece of text will have a unique file name.

(2) He will incorporate plots of data where relevant.

(3) He will perform a composition pass on the completed text using a dual-screen composition technique for putting the pieces together.

(4) He will perform formatting and text justification to set up the text report pages in the correct format.

(5) He will add the report body to the summary he created earlier and print a copy, as well as saving the entire report in the updated file.

(6) He can edit the printed report and have his secretary make the corrections.

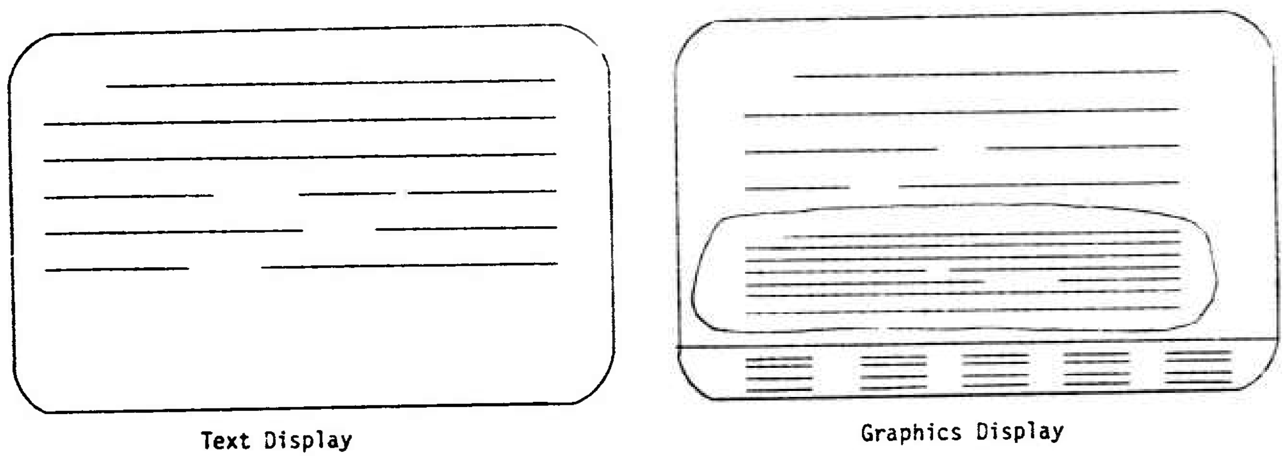
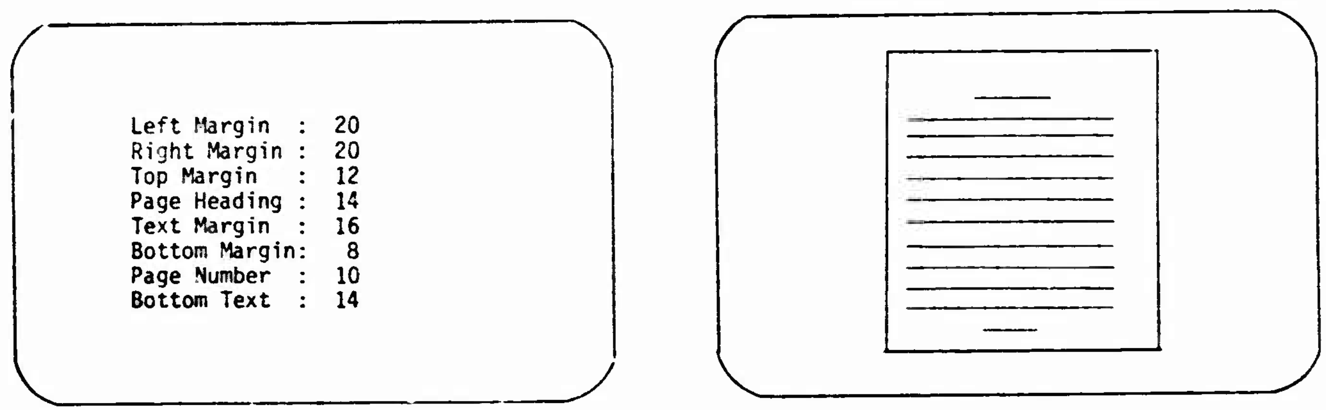


Figure 3-17. Two-Screen Composition with Graphics Selection



Text Disp?y

Graphics Display

Figure 3-18. Setting Up Page Format for Report

The secretary begins by creating text in conveniently defined pieces; Figure 3-15 shows the report creation in progress, after he has given the command,

ENTERTEXT PART1 XEQ.

After he has created all the pieces, the analyst is ready to start assembling them into a unified whole. He enters the command,

COMPOSE_TEXT LASTHALF XEQ,

where LASTHALF is the name of a new file for the completed text.

The 2-screen composition proceeds in the following manner: pieces of the report from the various files are brought in and displayed on the graphics screen. As the analyst determines they are ready, he sends them to the text screen, where they are added to the composite file, LASTHALF, as shown in Figure 3-16. Each time a file is sent for inclusion in the text, the system returns and requests the file name of the next piece to bring in the graphics screen. If the analyst makes an error or changes his mind about the order of the pieces, he is able to scroll the text backward and forward, make insertions or deletions, and change the order of various parts.

Let us assume that while the analyst is composing the text, he remembers a writeup in another file that may be relevant to this report. He enters the command,

GETFILE REPORT184 GRAPH XEQ,

and searches through various pages of the text until he finds what he is looking for. Using his data pencil and tablet, he circles one of the paragraphs in the text on the graphics screen. He then selects the menu item, TRANSFER_TO_TEXT_DISP (as shown in Figure 3-17) and causes this paragraph to be added on the text display following the last piece of previous material.

Once the text has been assembled into a sequential order in a single file, all that remains is setting up the report in page format. The analyst accomplishes this by entering the command,

GENERATE_REPORTHALF XEQ.

The system has various default values for page headings and margins, however, it confirms these with the user before beginning to process the text file. As shown in Figure 3-18, it does this by listing the margin and page top and bottom settings on the text display and drawing a page layout on the graphics terminal. As the user changes any of the settings, the drawing is changed to reflect the change.

Other page parameters that are requested are initial page number (the text will begin with page 2, since the summary is page 1) and page headings. When the system has all of the necessary information, it justifies the text into the proper page format. The analyst completes the report by giving the command,

APPEND_FILE LASTHALF REPORT259 XEQ.

He also enters,

```
PUTFILE REPORT259 TEXT
PRINTFILE REPORT259  XEQ.
```

Removing the completed report from the printer/plotter, the analyst has now solved his problem and produced finished intelligence. Using the User Communications facility, he has managed to do this in a much shorter time than would have been possible using previous methods.

4.0 USER COMMUNICATIONS - A SYSTEM VIEWPOINT

The previous section explored how an analyst user might react to the capabilities and facilities that are envisioned for the User Communications Interface. In this section, we present some of the problems faced by the designers and some considerations that influenced the form the design now has. Using essentially the same format as before, we shall examine first the environment and then the user language.

4.1 The System Environment

Since the period of the initial study to determine means of providing an on-line access capability for STIS, the project team have had as a goal the development of a user interface that would be useful, that would be usable, and that would be used. It follows that we will produce a system that is used, if we are successful in designing one that meets the criteria of usefulness and usability.

To be useful a system must meet specific needs and, hopefully, meet them better than any other available system. Although there are problems inherent in specifying the requirements to be met by something that doesn't exist (e.g., like convincing a horse farmer of his requirements for a tractor he has never seen), the User Communications project team has discussed analytical and information processing requirements with a cross-section of FTD analysts (see Section 2.2.4). As distinguished

from the issue of usefulness covered in Section 2, this section focuses on usability.

4.1.1 The Man-Machine Interface.

4.1.1.1 Focus of the Interface. Systems built in the past that involved a man-machine dialog too often concentrated on the machine functions at the expense of the person. Such systems were mastered only by very determined or very desperate users. The current User Communications design is an attempt to give the proper amount of thought and planning to the human side of the interface, i.e., the human information processing requirements. The system, while providing special help to users who have no previous computer experience, is intended to give all users those resources and aids that can maximize their effective use of the system.

Our underlying assumption is that FTD production analysts have problem solving tasks to perform, requiring data from existing computerized data bases -- e.g. STIS, analyzed sensor data, etc. -- access to automated analytical tools -- e.g., trajectory analysis programs -- access to graphics capabilities to plot data -- and access to automated report generation facilities to produce finished intelligence. The challenge in the User Communications development is to provide the kinds of system support facilities that will most closely match the analysts' needs. An early goal was to provide a system that would be self-teaching, i.e., would provide on-line tutorials and instruction sequences such as those

presented in Section 3. The primary concern, however, was that the design should reflect an understanding of how humans solve problems, especially in an interactive situation, and of how a computerized system should support this activity. Specifically, the user interface should help the analyst in stating his problem, in planning the solution, and in the execution of the plans.

4.1.1.2 Human Behavior in Problem Solving Tasks. The results of a great deal of experimental study lead to the conclusion that humans have two distinct modes of information processing. These modes can be identified to some extent with the left and right hemispheres of the brain. Although there is considerable overlap, certain kinds of processing are specific to one hemisphere or the other. In most persons, linguistic functions, mathematical ability, and analysis (the fine detailed structure), are left hemisphere functions, while the right hemisphere excels in recognition of shapes, imagery, and musical patterns (the integrated or holistic aspect of a musical selection). Bogen believes there is a dual representation, one in each hemisphere, for all information (5). He sees the most successful educational techniques as those which attempt to maintain a proper balance between the specialized processing capabilities of the two hemispheres. Wittrock shows how memory can be enhanced by imagery associations, a right hemisphere function (6).

For our purposes here, there are two points to be made from the preceding discussion. The first is that not only education but all other intellectual activities as well (including various kinds of problem solving) should take into account the two distinct aspects of human information processing. The second point is that in humans one hemisphere of the brain tends to dominate, producing some individuals who seem to be "analytical" and others who appear more "intuitive" in their approach to a given task. Any human-oriented computer system should be tailored to both approaches (e.g., this implies that there might be alternative structures for on-line tutorials).

A second significant finding from Experimental Psychology relates to human memory faculties. People have both a long term and a short term memory, where the long term memory stores information that may be recalled for an extended period of time and the short term one is used for momentary information processing assistance. It has been observed that humans can retain, on the average, a maximum of seven items (words, numbers, objects, symbols, etc.) in the short term memory. Applied to problem solving, this implies that, when the number of information items a human is required to deal with at the same time exceeds three or four, the human requires outside assistance.

A final human factors consideration is that people usually approach a complex problem solving situation having a general, but not a specific, plan for the solution of the problem. As they get into the problem,

they discover the need to solve several aspects of the problem which may not have occurred to them previously, in order to attack the more general problem. Planning, then, is an important element in problem solving; it includes defining the dimensions of the problem, identifying subgoals, discovering order dependencies, and deciding on one or more possible strategies for the solution.

Human information processing considerations provide considerable guidance as to the kinds of computer support needed for the on-line problem solving situation. First of all we observe that a strictly textual (i.e., verbal) approach to man-machine interaction is inadequate, because it emphasizes processing by the left brain hemisphere and the right hemisphere's contribution is essentially ignored. Graphics provides the missing element. Since the user can see elements of his problem and then connect them visually in various ways on the graphics screen, the intuitive and generalizing faculties of the right brain hemisphere can come into play and identify larger information patterns and more abstract aspects of the problem.

Moreover, Martin(7) says, "Graphics terminals have a remarkable appeal to the intelligent user, partly because their speed is close to the speed at which an intelligent person can absorb information. The typewriter-like terminal is frustratingly slow for most applications other than programming". To this Carlson and Sutton add, "A graphics terminal is essential to interactive problem solving in order to support a variety of data presentation techniques' (8).

An on-line system, using graphics as needed, should aid the user's short term memory in several ways: it should provide easy use of "boxes", connectors, and text. "Boxes" are identifiable and conveniently accessible places to store facts, elements, intermediate results, etc. Connectors, sometimes visual (e.g., graphics) and sometimes implicit, (i.e., maintained by the system), are links between various pieces of a problem (i.e., "boxes") that the user is working on; a plan in the process of formulation might use connectors. Text is messages and reminders the user makes for himself in the course of a work session so that he can easily recall: (1) why he made certain decisions, (2) how he arrived at a conclusion or result, (3) parts of a problem still needing attention, (4) where to find or how to identify missing elements. Text, boxes, and connectors can all be saved by the user so that he can recreate his strategy in attacking a problem.

Up to this point the most obvious requirement for graphics in the current application has been ignored -- information elements in intelligence are often so inter-related that the user needs special help in tracing them through. For example, a STIS node may have more than one logical predecessor (i.e., inverse relation); a user in data base navigation mode may not know how he got to a given node. By leaving "tracks" of individual STIS node accesses through the use of interactive graphics, the user is able to define a unique access path to the element he is interested in. Such tracks can also be saved by the user for future retrievals.

4.1.2 The Hardware Support System. Versatile terminal support and interactive graphics can be provided through many possible combinations of hardware and software. The particular combination assumed early in the period of the User Communications Interface global design was a medium size minicomputer configuration. The minicomputer hardware would include a refresh type graphics display, a display processor, and a graphics tablet or light pen. Graphics software would be available at the level of the user program and/or through calls to the minicomputer operating system. Such a combination of hardware and software is available in several current minicomputer packages.

As discussed in Section 3 and described in detail in Appendix A, a baseline configuration for the User Communications breadboard system would include the following hardware components:

- typewriter input keyboard;
- CRT display for typed information and system responses;
- hard copy device for text;
- CRT graphics display (refresh type) with display processor and refresh memory;
- hard copy device (plotter) for graphics;
- graphics tablet and pencil;
- local on-line storage (e.g., disk) for text and graphics.

An alternative to the pre-packaged minicomputer intelligent graphics terminal approach is that of a minicomputer-based system in which some peripherals may be interfaced by means of microprocessors. Since microprocessors would remove much of the I/O control burden, the minicomputer could be expanded to support multiple users in a timesharing environment. A complex of peripherals for one user would constitute a work station; several user stations could be attached to a single minicomputer, which in turn would have communication links to the mainframe computer on which the S&T data management system (i.e., STIS) resides. The possibility that the FY83 ADP concept will be implemented at FTD makes it attractive to reconsider User Communications hardware in favor of the single mini, multiple user station configuration.

Since the breadboard system is developmental in nature and FTD is currently in the process of elaborating the ADP concept discussed in Section 2.2, it is useful to think of the breadboard hardware as a precursor to the eventual User Communications hardware configuration. The breadboard, while not providing all of the hardware functions, should embody the same philosophy as production versions of User Communications hardware; for example, the breadboard User Communications might have a minicomputer supporting two user stations, while in the FY83 version of a similar mini might handle sixteen to twenty users. Based on our present understanding of User Communications requirements, we have surveyed currently available hardware and have produced two levels of baseline hardware support (a breadboard baseline and an FY83 baseline); these are presented in Appendix A.

Tentative breadboard hardware, consisting of a minicomputer and two user stations, is shown in Figure 4-1. The User Communications system is intended to allow the user to interact with his data in a variety of ways. While early versions of User Communications will require hardware support primarily for text and for graphics, it is not unreasonable to expect that future User Communications systems will require the use of video also in order to provide for computer produced reports which contain photographs and other types of intelligence imagery. At least the following kinds of hardware devices are in the initial User Communications Systems:

- typewriter input keyboard;
- CRT display for typed information and system responses;
- hard copy device for text;
- CRT graphics display (refresh type) with display processor and refresh memory;
- hard copy device (plotter) for graphics;
- graphics tablet and pencil;
- local on-line storage (e.g., disk) for text and graphics.

A typical user was introduced to such a system in Section 3.1.

The decision to design the user communications interface around a multi-station minicomputer resulted in some substantial benefits. Much of the user interface software can now be implemented on the minicomputer itself, rather than on the data management machine. Some of the virtues of this design are:

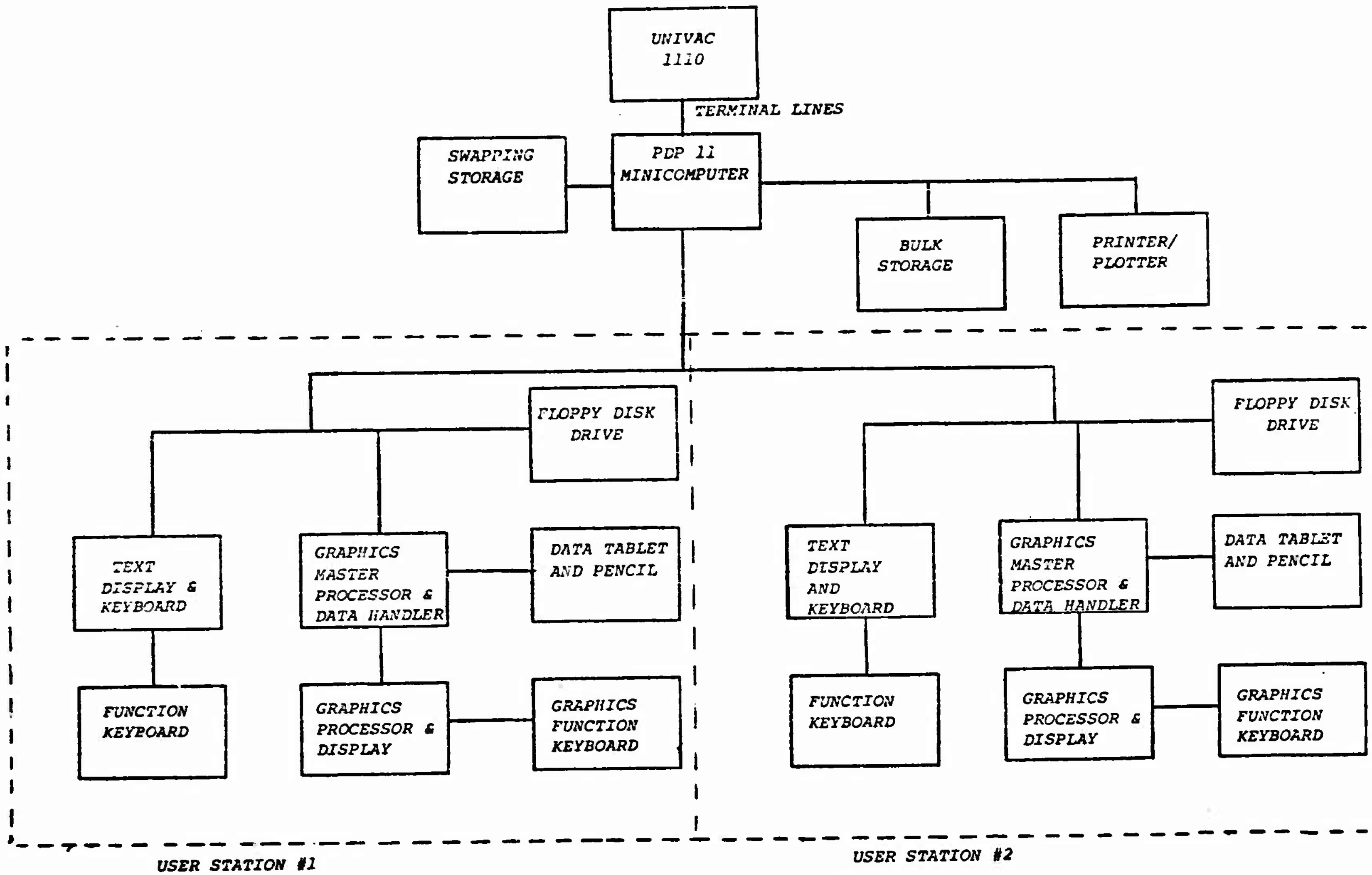


Figure 4-1. Tentative Breadboard Hardware Configuration

- The minicomputer will relieve the mainframe of nearly all the User Communications peripheral support;
- The minicomputer will off-load the processing requirements for user communications except for data management tasks and possibly other user programs residing on the mainframe. At the same time it will provide better response time for users;
- The minicomputer system may outlive the data management machine at FTD; though the data management system (i.e., STIS) may have to be implemented on a new machine, the User Communications software (and hardware) remains intact.
- The minicomputer configuration may resemble other such configurations (having other functions) to be developed in the future at FTD. They would share the same communications protocols and participate in a larger network of processors in the total ADP system.

Our original machine independent software design concept of a Conceptual Machine supporting a Dialog Supervisor and a Compiler/Interpreter is still valid, and is as straightforward to implement on the minicomputer as on the mainframe processor (see references 2, 3, and 4).

4.2 The User Language

4.2.1 <u>Philosophy Underlying the Language Design</u>. While the language for User Communications must support STIS access and update, it should provide other capabilities as well. Analysts who require access to the Scientific and Technical data base often need to use this data in the

context of a larger task. Data values need to be compared, combined, or operated on by a range of functions. A substantial amount of an analyst's time is dedicated to writing technical reports. The user interface should provide facilities for this, since in many cases values could be filled in directly from the data base or from routines provided by or constructed from user language commands.

Given that analyst activities are task-oriented, the ideal user language should match those tasks. Our design is based on the concept of language commands in the form of user functions. A problem solving activity is a form of goal oriented behavior, in which a major task is composed of subtasks. A subtask may be thought of as a sequence of related functions. By designing a language having an extensive repertoire of functions, we are providing the analyst with the tools he needs to perform (primarily on-line) most tasks requiring scientific and technical information stored in the data base. From a single user interface he will be able to examine (and if necessary, change) his data, perform necessary calculations, and construct a report showing his results.

The ultimate success and usefulness of man/machine information systems depends to a very great extent on the degree to which considerations of human information processing are applied to the design. IBM's Job Control Language for OS/360, 370 systems is a well-known negative example. While it is not necessary for the user to think of the system/machine as another human, everything about the system should feel natural and comfortable to him and should be easy for him to learn and adapt to. The system should be totally biased toward doing things the human way. In

this manner the communication potential between the human and his system can be maximized.

Applied to the User Communications language design, human information processing (HIP) considerations provide important design constraints. The language should be a compatible element of a total user-oriented environment. It should be functional and unobtrusive and should help to immerse him totally in his task. In practical terms, this implies that commands should neither be so verbose as to get in his way nor so abbreviated as to be ambiguous or to have questionable mnemonic value. The language and environment should be such that the user can easily move from one task (or information seeking mode) to another. Handling of error conditions in use of the language must also reflect HIP considerations. The user language should also be adaptive to the user, providing extensive guidance and prompting to a new user while allowing considerable freedom and flexibility to the more experienced one.

Our approach to the user language is based on two premises:

(1) the communication medium should be optimized to the specific set of tasks in the FTD production environment;

(2) the form of communication should take into account the background (training, skills, abilities, verbal habits, etc.) of the set of users.

Given that the User Communications interface is intended for use by S&T intelligence analysts, the users' information needs can be thought of in terms of a set of functions, which correlate with the task and subtask

elements and with data elements in the STIS data base. Our approach, then, is to cast the user language form in a functional mold. Intended to be used in a manner analogous to the electronic calculator, the function mechanism will be implemented in two complementary ways:

(1) a special function keyboard will be provided with function keys grouped according to functional subtasks;

(2) a "virtual" function keyboard containing functions as selectable "menu" items will be implemented using graphics and a data tablet and pencil to activate a given function (many users find a data tablet less fatiguing than a light pen).

In either representation the functions form part of a larger user-oriented environment based on the user station concept. A set of reserved keys are available to be assigned to user-defined functions.

The design fosters the impression on the non-computer oriented user that he is interacting with a "friendly" system that is optimized to his task. The system provides extensive tutorials and HELP functions, ability to tailor the interaction to the amount of user experience, e.g., adjustable verbosity, and graceful handling of error situations.

Several sets of command types have been defined in earlier phases of User Communications development. Some of these are:

- structural data base accessing and store functions;
- data base "navigational" functions;
- mathematical functions;

- editing functions;
- report generation functions;
- simple graphics (plot) functions;
- system communication functions;
- commands to permit user-defined functions, thus extending the language.

Examples of user commands, as well as users of these commands in a problem solving environment were illustrated in Section 3.

4.2.2 Future Language Enhancements. In our view, the user language design will provide the essential tools for user analysts to perform their tasks. Extensibility, in the form of definition of new functions by the user gives the language practically unlimited power for the intended applications.

While the user language has the capability, in principle as well as in fact, to specify almost any action to be executed, the language shares with other computer oriented languages the characteristic that some kinds of execution specification are more convenient than others. There are at least two directions in which the user language will expand to provide a form more convenient to specialized users:

(1) An arithmetic expression subset;

(2) A query statement subset.

In the first instance, the language will provide the facility for experienced programmers to input A + B = C; instead of ADD A B MOVE C XEQ. (Note that the input effort is exactly the same, since the first

method involves 6 strokes on the alphanumeric keyboard and the second involves 3 alphanumeric strokes for the variables A, B, C, and 3 function key strokes for commands). The second expansion would allow a user to say, "FOR ICBM DISPLAY VALUES WHEN ATTRIBUTE_NAME = LENGTH AND DOB ≥ 700501, rather than having to define a new function composed of existing functions to perform the same retrieval.

That our language approach is sound is evidenced by the ease with which we may add translators for these two expansions of the language. The language functions provide the primitives for both kinds of translation. The arithmetic expression capability is the simplest, consisting primarily of a program which takes as input an <u>infix</u> arithmetic format and translates it into the <u>prefix</u> functional form. The principal difficulty is deciding <u>which</u> arithmetic expressional form to use (FORTRAN, PL/1, APL, etc.).

The difficulty in developing a query statement subset of the user language is one degree (as opposed to the arithmetic expression capability) rather than of kind. A translator for a query statement requires a more complex program, since a single statement may map into a relatively large number of functions (i.e., a one to many mapping, as contrasted with the one to one mapping for arithmetic expressions). Similarly, it is more difficult to decide on (or agree on) the form a query statement should take. Natural English is undesirable for the reasons given in References (1) and (9).

Rather the query statement form should be "English-like" but highly structured; the syntax should expand into completely predictable forms (for English speakers), and the syntactic rules of the queries should be easy to learn and use.

4.2.3 Difficulties in Designing Computer Language for Non-Computer Oriented Users. Given that our user language design is geared to the inexperienced, as well as the experienced computer user, a number of problems immediately surface. In the current application, some of the problems are of a general nature while others relate to specific features of STIS.

4.2.3.1 General Language Problems. A persistent problem in the language design has been the difficulty of implementing advanced programming concepts without requiring a language user to understand them. The following sections discuss these and indicate solutions selected for the breadboard implementation, which are described in References 2, 3, and 4.

4.2.3.1.1 Saving and Manipulating User Variables. There are problems relating to user variables and the values (i.e., bindings) they assume. The ideal for a non-programmer is a language in which there is no explicit movement of data from one storage location to another; all values are handled by the language system. Query languages exhibit a range of philosophies concerning user variables and temporary storage of retrieved data. In the simplest cases, the systems make no provision for storage of or calculations on retrieved values; they are simply

displayed or written out on the user terminal. In these systems, no grouping of queries (chained retrieval with calculations on values) is possible. The next level of sophistication is query systems containing an unnamed work area. Data retrieval causes sets of data elements to be moved to a system work area, where other operations (e.g., ordering, maximum, minimum, etc.) can be applied to them by a subsequent query. Finally, flexible query systems (such as ADABAS and RAMIS) not only have work areas for an entire record, but they also support user variables (calculation of totals, counts, etc.) and accumulations on named data fields within the record.

Students in a first programming course have to grasp the concept that program values have to reside somewhere in a memory location, (i.e., a "box"), of the computer. With the wide acceptance and use or pocket calculators, the idea of using special memory locations to store values and intermediate results is becoming more generally understood. On this basis our design assumed that users could learn to use specially designated function keys as memory locations for values required in language functions. It was recognized that the experienced programmer would want to use symbolic variables as well as function key memories. If the language were to allow this, then it must provide a unified framework that included both. The difficulty arises in foreseeing whether the non-programmer will easily adjust to <u>symbolic</u> memory names, i.e., a reference to the <u>name</u> of a function key memory location causes

its value to be retrieved automatically. The Recall function of the calculator (e.g., RCL, MR, etc.) is bypassed; however, under various conditions, storage must be explicit.

For example, a typical calculator sequence is RECALL 1
ADD
RECALL 2
STORE 3.

The same computation is performed in the user language by the following sequence of 6 function key strokes:

```
ADD MEMORY1 MEMORY2
MOVE MEMORY3
XEQ
```

or even perhaps

```
ADD MEMORY 1 MEMORY2 MEMORY3
XEQ.
```

4.2.3.1.2 Delimiting Symbolic Variables. If the user language supports symbolic variables, then another source of possible confusion is the character string data type. Using quotes to delimit strings (the convention followed in many programming languages), "MEMORY5" might be a value stored in MEMORY5.

4.2.3.1.3 Implementing Functions. The strategy for implementing functions in the language is another issue. Out of concern for the naive user, it is desirable to keep the function mechanism as simple as possible. The functional arguments would follow the function name,

with spaces used for separation; no parentheses, commas or other additonal complexity would be required. The problem with this simple language syntax is that it does not allow for composition of functions. With the appropriate punctuation, it is possible to execute a sequence such as:

```
AVERAGE (MAX(A, B, C), MIN (D, E, F))
XEQ,
```

rather than having to produce the same result sequentially:

```
MAX A B C
MOVE MEMORY1
MIN D E F
MOVE MEMORY3
AVERAGE MEMORY1  MEMORY3
XEQ.
```

Without parentheses, there is no way for a compiler or an interpreter to deal with function composition, since the input is ambiguous as to where one function ends and the next one starts.

Another conceptual problem relating to functions has to do with user defined functions. If user functions are allowed to have arguments (presumably a user will want to pass values to some of the functions he defines), will the user understand the difference between formal parameters, (i.e., "dummy" variables in FORTRAN terms) in the function definition and real arguments at the time the function is used?

4.2.3.1.4 Defining the Scope of Variables. A final computer language concept that might cause difficulty relates to the scope of variables. Once again the problem is that what is understandable to the novice will not be satisfactory to the experienced programmer. The issue here

its value to be retrieved automatically. The Recall function of the calculator (e.g., RCL, MR, etc.) is bypassed; however, under various conditions, storage must be explicit.

For example, a typical calculator sequence is RECALL 1
ADD
RECALL 2
STORE 3.

The same computation is performed in the user language by the following sequence of 6 function key strokes:

```
ADD MEMORY1 MEMORY2
MOVE MEMORY3
XEQ
```

or even perhaps

```
ADD MEMORY 1 MEMORY2 MEMORY3
XEQ.
```

4.2.3.1.2 Delimiting Symbolic Variables. If the user language supports symbolic variables, then another source of possible confusion is the character string data type. Using quotes to delimit strings (the convention followed in many programming languages), "MEMORY5" might be a value stored in MEMORY5.

4.2.3.1.3 Implementing Functions. The strategy for implementing functions in the language is another issue. Out of concern for the naive user, it is desirable to keep the function mechanism as simple as possible. The functional arguments would follow the function name,

is whether the user language variables should be global or local (or both) in scope. That is, if a variable (e.g., a memory) is manipulated internally to a function, what value should the variable have once execution is resumed in the main body of the program, after the function call? There are convincing arguments on both sides of the question, since either decision may make possible situations in which unexpected things could happen in the user's programs.

4.2.3.1.5 Defining Loops. Besides the conceptual issues, there are problems of a more practical nature. The language obviously must provide an iteration (looping) capability, since an analyst will need to step through a sequence of data base nodes and examine or collect the values of various attributes.

The "LABEL and GOTO" approach was abandoned as being potentially dangerous, since it requires considerable care to avoid infinite loops, incorrectly initialized terminating conditions, and generally bad logic in the layout of language functions. The strategy adopted is to provide a special kit for assembling loops, including a small set of special loop type functions that the user could define. The loop testing and branching logic will be handled by system-generated code that is invisible to the user. Special test functions in the user language will provide the capability to determine when the looping acitivity should terminate what should be done with a given value, and similar program considerations.

4.2.3.2 STIS-Related Functions. Intelligence data in general and S&T information in particular is highly complex in nature, and an information system to support S&T analysis of necessity reflects this complexity

of the real world. Names of real-world objects and relations between them, as well as values of real-world parameters are things analysts deal with. In addition, the realm of intelligence requires dimensions of labelling and control on facts that are not essential for other DBMS applications. The necessary complexity of the intelligence information base also increases the complexity of the user language:

(1) Names and values in the data base require special care for mapping into user language data types;

(2) The complexity of STIS attribute values requires a corresponding complex syntax in the user language functions that store, retrieve, modify, or delete these values.

The first problem is related to the fact that some names in the data base contain multiple words. Since blanks must be token delimiters in the user language (any other alternative would introduce unacceptable complexity into the language), multiple word data must be treated as character strings. The result is that the user will have to remember to enclose data element names in quotes.

More far reaching is the syntactic complexity of data base access commands. STIS has features that make it ideal for intelligence purposes; some of these are:

- Multiple values for attributes
- Array values
- Structured values
- Warnings attached to values

* Fact sequencing
* Fact Control Information (FCI) at the level of each value.

The sum total of all these features is an attribute value model that is very highly structured. User language functions that access attributes must be sufficiently precise to access the exact part of the value that is required. This involves schemes for indexing into arrays, sequencing through structures, looking at warnings, specifying pre-determined values in a field of FCI parameters. Several solutions have been developed, none of which goes very far in saving the inexperienced user from an undesirable level of syntactic detail for the STIS access functions. The availability of system default values (especially in the case of FCI information) does provide some relief, however, and the extensive HELP tutorials will give guidance when the user can't remember.

4.2.3.3 Special User Data Structures. Because of earlier uncertainty about the nature of the software and hardware environment for User Communications, the current design does not include the mechanisms for providing the "boxes", "connectors", and "text" capabilities, that were referred to in section 4.1 or the user language commands for creating, storing and accessing them. Now that the mini-computer based user station concept has solidified, these mechanisms and the corresponding user language functions can be developed.

4.2.3.4 Levels of Function. The difficulty in designing a language for non-programmers with application to the complexities of intelligence data has been evidenced by the specific problems mentioned

earlier. Focusing on a somewhat higher level of design objectives, we faced the problem of how to design a user language providing a multi-level capability. We wanted to provide significant language capabilities for virtually all levels of experience. The language design should be such that the novice could find his way along, using simple commands and simple versions of language functions, while the experienced programmer would find the language at least as powerful and as general as the programming language(s) he is competent in. Our interpretation of Reisner's results indicates that non-programmers eventually have to learn some programming concepts in order to realize the full spectrum of on-line language capabilities (10). The more difficult ideas have to be mastered. Nevertheless, the user language we have designed does contain "layering", so that a user is able to apply the level of computer sophistication that he has consciously or unwittingly acquired.

4.2.4 Motivation for a User Communications Breadboard System. In spite of the difficulties of defining a user language, we believe that workable solutions have been incorporated into a breadboard design, that the resulting language capability is impressive, and will provide the foundation for development into an important FTD resource.

However -- although careful consideration has been given to all the design issues mentioned in the preceding section as well as other issues not discussed in this part -- it would be naive to assume that our design decisions were correct in every detail and that users will not find improvements and changes they would like to have made in the language and

the system. It is clear that some period of analyst hands-on experience with the User Communications System will be necessary in order to more specifically tailor the system to FTD user requirements and, in general, to provide a 'shakedown' phase for the system.

For these reasons, the system designers prefer to introduce the User Communications System into the FTD environment in the form of a breadboard or experimental model, which will provide a test of the essential design features and allow for modification and improvement without major redesign or duplication of the development effort.

As discussed in Section 4.1, our concern focuses on the usability and usefulness of the system, knowing that it will be used by the FTD analyst community if it is easy to use (usability) and provides him with automated tools that assist him in performing his analytical tasks (usefulness).

The motivation for the User Communications breadboard is thus to provide an opportunity for FTD analysis to test and evaluate the usability and usefulness of the system before it is cast in concrete. This shakedown period of hands-on evaluation by FTD analysis will allow tests of usability in terms of human engineering, effectiveness of tutorial, error diagnostic, and help functions, suitability of commands and command names, approach to function definition, and so forth.

The only proper test of usefulness of an entirely new capability is a period of hands-on evaluation by actual users, since no previous experience with such a capability exists at FTD. It is well known

that user requirements change dynamically as user analytical and information seeking behavior is modified by the novel experience of interactive computer access. Thus an evaluation of the usefulness of an automated system for supporting users in their analytical and information processing activities must be performed by users interacting with the system in their working environment. The transaction monitor built into the User Communications breadboard will show which user language functions are used and which are not, user evaluative comments will assist in determining why commands are used or not used, and which additional commands would be useful in their work.

Because the breadboard system will be programmed as much as possible in a higher level language (LISP), it will be relatively simple to modify, and some modifications may be made and tested during the evaluation period. Moreover, the utilization of a higher level language will to the extent possible insulate User Communications against the effects of new hardware procured by FTD in the 1980 timeframe. However, frequently used functions can be optimized by coding in assembly language as function calls within the structure of the higher level language.

In any case, the modifications and enhancements required for the future operational User Communications System tailored to the needs of the FTD analyst community can be introduced without major design overhauls and duplication of development efforts due to the implementation of a breadboard as a first operational step.

Appendix A -- Baseline Hardware Systems

The term "baseline" is understood here as meaning basic, minimal, and essential. In other words the individual hardware items selected represent essential capabilities, and the corresponding items to be acquired for User Communications breadboard terminal support should not be inferior to the devices described here in capabilities, power, or flexibility.

It is recognized that in the case of most of the devices selected, there are acceptable alternatives from other manufacturers. Furthermore, due to the rapidly changing technology, as soon as the time is ripe to acquire the equipment, new products will have become available having substantially superior capabilities and lower price than the items described here. Prices quoted for hardware are purchase prices, and do not include possible discounts, such as OEM discounts.

The Baseline Breadboard System

The configuration chosen for the User Communications breadboard system is based on certain assumptions:

(1) A minicomputer will support two user stations in timesharing mode.

(2) Realities of the Univac 1110 operating environment at FTD dictate that, for the present, the link between the U1110 and the minicomputer will have to consist of demand terminal lines.

(3) The major part of the User Communications software will reside on the minicomputer. The minicomputer system will therefore require sufficient resources to guarantee flexibility for program development, as well as extensive support and fast response time for on-line users.

The Minicomputer

For compatibility with the proposed ADP concept for FTD FY83, hardware from Digital Equipment Corporation is indicated. The timesharing capability being a major consideration, the minicomputer mainframe should be at least a PDP 11/45, and preferably a PDP 11/70. It should include 64K of memory and the memory management option. Floating point hardware is desirable. While there are a relatively large number of configurations available, based on the kinds of memory included and what other options are provided, the average prices are in the range of $40,000.00 for the PDP 11/45 and $65,000.00 for the PDP 11/70.

Mass Storage

The minicomputer requires a magnetic disk pack storage system, including a drive unit and a controller. There are several manufacturers with comparable equipment; the choice will ultimately have to be made on the basis of various considerations. At least 75 to 80 megabytes of storage should be available per disk pack. Several vendors offer small packs (10 data surfaces) with capacities in this range. To our knowledge DEC does not yet provide packs with this storage density. The relatively small price difference between drives for the small and large packs may make it attractive to consider the large pack drive.

The disk drive controller is an important consideration. Most controllers will support multiple drives. An attractive feature in a controller is a programmable microprocessor, which provides greater flexibility in data buffering and error checking and correction.

Current practice is for disk pack system vendors to package OEM drives from other manufacturers with their own controllers. Telefile, Diva, and Caelus are currently offering very attractive systems. Diva's package is offered for approximately $16,000.

Swapping Storage

Timesharing requirements dictate the need for a very fast auxiliary storage. Conventionally this function has been provided by magnetic drums or by fixed-head magnetic disks (together with a controller). Acceptable fixed head disks are available from a number of manufacturers. A reliable system should be available for approximately $18,000.00 to $20,000.00.

A newer technology may make fixed head disks obsolete in the near future for many applications. Texas Instruments is already marketing prototype Charge-Coupled Device (CCD) MOS (Metal Oxide Seimiconductor) memories with 65,536 bit capacity on a single 16 pin IC (integrated circuit) chip. Other semiconductor manufacturers are developing similar devices. If the price comes down in 1978 as expected to about $13.00 per chip, these devices will become an attractive alternative to a fixed head disk; a memory and controller could probably be built for $8000.00 or less.

Printer/Plotter

Because of the expense of other system components, an electrostatic printer/plotter was selected as the means of providing system printing, as well as text and graphics hard copy. While not ideal in some respects, such a device should provide the minimal capabilities required in the breadboard application. An electrostatic system was chosen on the basis of performance versus cost considerations. Versatec's Model D1200A, with a PDP 11 compatible controller and the simultaneous print/plot option, sells for about $10,750. This model produces excellent print quality, and with scan conversion software or hardware, it can also make copies of the graphics display. While a separate and probably larger plotter might be desirable, it was judged to be too expensive an item for this stage of User Communications implementation.

The User Stations

Each of the following items are present in each user station.

Text Terminal

A typewriter type terminal with a CRT display was chosen on the basis of its power to input and manipulate text data. A built-in microprocessor gives the device significantly more capability than the conventional terminal has (e.g. ability to save text "pages" in local buffers for purposes of scrolling, powerful local editing features, etc.). A Beehive Medical Electronics, Inc. model B500 was selected, the options being a 15" screen, programmable keyboard, standard and programmable character generation and expanded memory.

The cost of this configuration is approximately $3500.

Graphics Terminal

A raster scan type graphics system was chosen over the stroke generation type primarily on the basis of its compatibility with video input and its significantly lower cost. Graphics systems require more hardware, and special care must be taken to insure that 1) data paths and mass storage must allow for sufficiently high data rates to keep up with the desired speed of the graphics displays, and 2) the large number of I/O operations to support graphics must not be allowed to burden the minicomputer, that is supporting timesharing activities. The graphics system selected has several components.

Graphics Processor

The heart of graphics system is a Genesco Computers GCT-3000. It contains a Programmable Graphics Processor (PGP), a Video control unit, MOS memory for refresh, a 17" Monitor, a PDP/11 interface, and a cabinet and power supply. Also included, if desired, is a data tablet with an RS-232 interface. The total cost is approximately $13,780. If the data tablet were to be purchased elsewhere, the resulting configuration would cost $9,980.

Support Processor

Because of the need to insulate the minicomputer mainframe from the graphics I/O activity, it will be necessary to provide an additional processor to assume that support. In addition the processor would handle special calculations required to provide zooming capabilities and other highly dynamic aspects of graphics. Graphics specifications have not yet been

worked out in detail, so that the exact nature of the support processor is not known. A microprocessor, such as an INTEL 8080A or an LSI 11, may be adequate; if not another DEC processor such as a PDP 11/04 or 05 or 10, or 34 should be considered. If the processor is another DEC computer, a Unibus to Unibus interface will also be required. At this point in time a very rough estimate for the cost of the support processor is $6,000.

Floppy Disk

A diskette system will provide a low cost but reasonable capacity for direct user data storage. Many systems are available, including one from DEC. The diskette drive selected here has the advantage of using a highly reliable Shugart drive. The diskette drive and controller are available from Standard Logic Systems, Santa Ana, California, at a cost of approximately $2500.

PDP 11 Operating System

Two candidate systems providing time sharing capability on PDP 11 computers are DEC's RSX-11 and the UNIX system available from Western Electric. Because of its very elegant design, including a very powerful file system, a simple and general I/O system, and the ability of user processes (programs) to "spawn" and control sub-processes, the UNIX system is preferred for the User Communications application. In addition signigicant work has been done on system security under UNIX, and there already exist ARPA Network type software interfaces.

The FTD FY83 Baseline System

The assumptions concerning this system are slightly different:

1) A single minicomputer will support 16 to 20 user stations;
2) There will be multiple minis, each one with its array of users;
3) The minicomputers will be interfaced to a system of processors, including the data management computer, by means of a network connection.

Many of our conclusions concerning this system have to be of a general nature, since it is not possible to foresee impact of new technology upon computer-related equipment. It is assumed that plotting devices will be provided as a resources to be shared by several users. The graphics may be more sophisticated to allow for video data.

The Minicomputer

At this point in time it appears that the DEC PDP 11/70 (or an improved equivalent model) is the best choice. Supporting more users, the computer will require significantly more swapping storage and a much larger mass storage. It is likely that mass storage of that time period will be significantly different from what is now available.

The User Station

The breadboard version of the user station is relatively complete in its concepts. Changes will be in the nature of improvements. Typewriter terminals and other kinds of peripherals will become faster and more "intelligent". A more specific specification for an FTD FY83 system should wait until the 1981-82 time frame.

REFERENCES

1. Operating Systems, Inc., STIS User Communications Interface: Functional Description, RADC-R-75-028, 31 December 1975.

2. Planning Research Corporation, FTD User Communications Interface Functional Description System/Subsystem, PRC Report No. R-2009, February, 1977.

3. Planning Research Corporation, FTD User Communications Interface: System/Subsystem Specification (External Subsystem Report), PRC Report No. R-1936, December, 1976.

4. Planning Research Corporation, FTD User Communications Interface, Program Specification, PRC Report No. R-2013, February, 1977.

5. Bogen, J. E., "Some Educational Aspects of Hemispheric Specialization", published in UCLA Educator, Volume 17, No. 2, Spring 1975, University of California, Los Angeles.

6. Wittrock, M.C., "The Generative Processes of Memory", UCLA Educator, Vol. 17, No. 2, Spring, 1975. University of California, Los Angeles.

7. Martin, J., Design of Man-Computer Dialogues, Prentice Hall, Englewood Cliffs, New Jersey, 1973.

8. Carlson, E.D., and Sutton, J.A., A Case Study of Non-Programmer Interactive Problem-Solving, IBM Research publication RJ 1382 (#21293) April 19, 1974.

9. Montgomery, C.A., "Is natural language an unnatural query language?" Proceedings of the ACM (Invited paper, Query Language Session): 1075-1078, August 1972.

10. Reisner, Phyllis, "Use of Experimentation as an Aid to Development of a Query Language", in IEEE Transactions on Software Engineering, Volume SE-3, No. 3, May 1977.